Sensing Structured Signals with Active and Ensemble Methods

by

John Lipor

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering: Systems)
in the University of Michigan
2017

Doctoral Committee:

Assistant Professor Laura Balzano, Chair
Associate Professor Jason J. Corso
Assistant Professor Branko Kerkez
Associate Professor Clayton Scott

John Lipor
lipor@umich.edu
ORCID iD: 0000-0002-0990-5493

To Ezra.

ACKNOWLEDGMENTS

I would like to begin by thanking my advisor, Professor Laura Balzano, for the many hours of time spent investing in me. During our time together, she has displayed immense patience through my "stupid" questions and trivial mistakes, fostering an environment of academic freedom that I hope to emulate with my students. I cannot overstate how much I have learned from her on how to approach problems, persevere through being stuck, and keep pushing until I understand every last detail. I am also thankful for her continuous support through family tragedy and joy, and I consider her a role model both professionally and personally.

I am grateful to every professor I had the chance to interact with during my time as a student. To Professor Barry Van Veen for being my first research advisor *and* academic great-grandfather. To Professor Mohamed-Slim Alouini for guiding me through my master's thesis. Thanks to my committee members for their valuable feedback on my work—Professor Clay Scott for answering my earliest questions on active learning, and Jason Corso for fielding later questions on the same topic. Thanks to Branko Kerkez and Don Scavia for being a wonderful pair of collaborators. My experience working with you has motivated me to continue seeking out interdisciplinary problems and played a major role in my obtaining my first "real" job. Thanks to Professors Alfred Hero, Raj Nadakuditi, Jeff Fessler, and Demos Teneketzis for your excellent teaching, advice, and for shaping the culture of our department. Lastly, I would like to thank Professor Roman Vershynin for his advice on a number of research questions.

Next, I would like to thank a number of my peers. First and foremost, David Hong and Dejiao Zhang for shaping the way I think through problems—I have learned so much from you two, both in terms of growing technically and being a more patient person, and I have been lucky to spend the last several years studying with you. Next, Matt Kvalheim for the hours spent discussing any and every mathematical topic. I would also like to thank Aniket Deshmukh and Julian Katz-Samuels for our great conversations on research, classes, and life. Finally, thanks to Brandon Wong for the hours spent hacking and building to make Perry run.

Finally, thank you to the people who have raised me, lived with me, and made me who I am. To my parents, who instilled in me the value of hard work for as long as I can remember and gave me the opportunities that led to my love of education. Most of all, thank you to my wife, Gina, for journeying with me from Saudi Arabia to Michigan and now to Portland, for always pushing me to do more than I think I can, and for putting up with a husband who will begin his first job at age 30.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF APPENDICES

ABSTRACT

Modern problems in signal processing and machine learning involve the analysis of data that is high-volume, high-dimensional, or both. In one example, scientists studying the environment must choose their set of measurements from an infinite set of possible sample locations. In another, performing inference on high-resolution images involves operating on vectors whose dimensionality is on the order of tens of thousands. To combat the challenges presented by these and other applications, researchers rely on two key features intrinsic to many large datasets. First, large volumes of data can often be accurately represented by a few key points, allowing for efficient processing, summary, and collection of data. Second, high-dimensional data often has low-dimensional intrinsic structure that can be leveraged for processing and storage. This thesis leverages these facts to develop and analyze algorithms capable of handling the challenges presented by modern data.

The first scenario considered in this thesis is that of monitoring regions of low oxygen concentration (hypoxia) in lakes via an autonomous robot. Tracking the spatial extent of such hypoxic regions is of great interest and importance to scientists studying the Great Lakes, but current systems rely heavily on hydrodynamic models and a very small number of measurements at predefined sample locations. Existing active learning algorithms minimize the samples required to determine the spatial extent but do not consider the distance traveled during the estimation procedure. We propose a novel active learning algorithm for tracking such regions that balances both the number of measurements taken and the distance traveled in estimating the boundary of the hypoxic zone.

The second scenario considered is learning a *union of subspaces* (UoS) model that best fits a given collection of points. This model can be viewed as a generalization of principal components

analysis (PCA) in which data vectors are drawn from one of several low-dimensional linear subspaces of the ambient space and has applications in image segmentation and object recognition. The problem of automatically sorting the data according to nearest subspace is known as *subspace clustering*, and existing unsupervised algorithms perform this task well in many situations. However, state-of-the-art algorithms do not fully leverage the problem geometry, and the resulting clustering errors are far from the best possible using the UoS model. We present two novel means of bridging this gap. We first present a method of incorporating semi-supervised information into existing unsupervised subspace clustering algorithms in the form of pairwise constraints between items. We next study an ensemble algorithm for unsupervised subspace clustering that functions by combining the outputs from many efficient but inaccurate base clusterings to achieve state-of-the-art performance. Finally, we perform the first principled study of model selection for subspace clustering, in which we define clustering quality metrics that do not rely on the ground truth and evaluate their ability to reliably predict clustering accuracy.

The contributions of this thesis demonstrate the applicability of tools from signal processing and machine learning to problems ranging from scientific exploration to computer vision. By utilizing inherent structure in the data, we develop algorithms that are efficient in terms of computational complexity and other realistic costs, making them truly practical for modern problems in data science.

CHAPTER 1

Introduction

1.1 Motivation

Throughout the last decade, we have witnessed a paradigm shift in which nearly every aspect of our lives is now measured, recorded, and processed with the goal of having modern life be "data driven" in nearly every domain. Step counters quantify our movements throughout the day, recommender systems track our buying, watching, and listening preferences, and self-driving cars instantaneously process the world around them. The task of the modern "data scientist" is to utilize tools from mathematics, statistics, and computer science to convert this trove of data into actionable information. However, due to the unprecedented magnitude of data—both in terms of volume and dimension—traditional techniques from these areas can no longer keep pace with the current demand. For this reason, researchers rely on the following two key facts that are intrinsic to many modern datasets.

1. Large volumes of data can often be *accurately represented by a few key points*, allowing for efficient processing, summary, and collection of data.

2. High-dimensional data often has *low-dimensional intrinsic structure* that can be leveraged for processing and storage.

The goal of this thesis is to leverage these two facts to develop and analyze algorithms capable of handling the challenges presented by modern data. One approach to seeking out the most salient data points comes from the area of *active learning*, where researchers hope to gain a clear understanding of which points are most informative through successively querying points and updating their data model. However, existing algorithms are either theoretically grounded but disconnected from practice or practically useful but lacking in theoretical justification. The active methods presented in this thesis aim to bridge this divide, providing both theoretical guarantees (or justification) and strong empirical performance on real data. Determining low-dimensional structure in data is also a widely-studied problem, with the most popular approach being that of

principal components analysis (PCA). The generalization of PCA to unions of subspaces is known as *subspace clustering*, and is a widely-studied problem in the signal processing community. However, existing algorithms do not utilize the full capacity of this rich model, achieving performance that is significantly worse than the best union-of-subspaces classifier on most real datasets. We present two avenues toward better leveraging of this model; the first incorporates actively-chosen pairwise constraints between items, and the second combines the results of many inaccurate base clusterings to form a global clustering that achieves state-of-the-art performance.

This thesis presents projects in three modern data processing scenarios, and in this chapter we discuss an overview of modern tools and results that are common across all projects. These algorithms rely on tools from the areas of active learning and high-dimensional probability, both to gain insight into problem structure and to perform rigorous analysis. The aim of this chapter is to present the reader with the vernacular of these projects and to provide a brief overview into the main content of the thesis.

1.2 Major Contributions

1.2.1 Quantile Search: An Active Learning Algorithm for Spatial Sampling

In Chapter 2, we present an active learning algorithm for binary classification in spatial sampling problems, where the sampling cost depends on both the number of samples taken and the distance traveled during the learning procedure. In contrast to traditional supervised learning, the active learning paradigm allows the user to query specific points in order to reduce the number of labeled examples required to learn a decision boundary. Motivated by the goal of estimating the spatial extent of low-oxygen regions in Lake Erie, our aim is to accurately estimate the boundary of such regions using *as little sampling time as possible*. Oxygen concentration is a strong indicator of the health of the Great Lakes [3] and the spatial extent of hypoxic regions is a topic of interest for researchers in the field [4]. Active learning algorithms have been proposed as a method of minimizing the number of samples required for similar tasks [5, 6]. However, the central basin has an area of roughly 11,000 km^2, making the total sampling time dependent on both the number of samples taken and the distance traveled throughout the estimation procedure. In this chapter, we present an algorithm referred to as *quantile search* for estimating the change point of a one-dimensional step function on the unit interval. Quantile search is an extension of greedy active learning and has the ability to balance the number of samples taken and distance traveled via a tuning parameter; at one extreme, quantile search is equivalent to binary search, while at the other it is essentially continuous sampling. We present algorithms to handle both noiseless and noisy measurements, as well as theoretical guarantees. We then show how the original estimation problem

can be transformed into a series of one-dimensional problems and demonstrate the time reduction achieved by our algorithm. Finally, we demonstrate the effectiveness of our algorithm in the real world by presenting results from experiments performed on Third Sister Lake in Ann Arbor, MI.

1.2.2 Active Learning for Subspace Clustering

In Chapter 3, we describe a method of incorporating actively queried human-provided constraints into subspace clustering algorithms. We consider the union of subspaces (UoS) model, in which we are given N points in $\mathbb{R}^D$ lying on a union of K subspaces of dimension $d < D$. This model has been shown to be applicable for a variety of real datasets, such as images of human faces under various lighting conditions [7], handwritten digits [8, 9], or objects under a variety of poses [10, 11]. Unsupervised subspace clustering algorithms achieve strong performance on these datasets but still fall short of the best possible performance under the UoS model. Moreover, in the case where a user wishes to correctly cluster *all* items in a database, including those that do not correctly fit the UoS model, some supervised input is certainly required. In the examples given, a human without expert knowledge could easily provide this input in the form of pairwise comparisons by answering questions such as whether two images depict the same face, digit, or object. The incorporation of such information into clustering algorithms is referred to as pairwise constrained clustering (PCC) and has been widely studied in recent years with strong results [1]. In [12], the authors note that incorporating poorly-chosen constraints can lead to an increase in clustering error, rather than a decrease as expected, since points constrained to be in the same cluster that are otherwise dissimilar can confound the PCC algorithm. Moreover, obtaining these comparisons may be expensive, as they require human input. For these reasons, researchers have turned to *active* query selection methods, in which constraints are intelligently selected based on a number of heuristics. While these algorithms provide major benefits over incorporating random constraints, they fail to leverage any underlying structure in the data. In this chapter, we present a PCC algorithm called SUPERPAC (SUbsPace clustERing with Pairwise Active Constraints) that takes into account the UoS geometry to attain state-of-the-art performance in PCC. We define and analyze a notion of relative margin that is specific to the UoS model. We then demonstrate the effectiveness of our algorithm on several benchmark datasets, showing that with a modest number of queries we see significant gains in clustering performance compared to existing algorithms. To the best of our knowledge, we are the first to consider the use of active learning in the context of subspace clustering.

1.2.3 Ensemble Methods in Subspace Clustering

In Chapter 4, we present a novel approach to the unsupervised subspace clustering problem that leverages ensembles of the K-subspaces (KSS) algorithm [13, 14, 15] via the evidence accumulation

clustering framework [16]. In the context of classification, ensemble methods combine many "weak" learning algorithms into a single classifier to achieve excellent classification performance. The best-known of these algorithms are Random Forests [17] and AdaBoost [18]. In a comprehensive study of 179 classifiers and 121 datasets, the authors of [19] found Random Forests to emerge as the best general-purpose classifier.[1] The goal of this chapter is to leverage the success of ensemble methods in classification for the case of subspace clustering. While the optimal KSS solution corresponds to near-perfect clustering performance, the alternating optimization approach used typically performs poorly in practice, even when the best result from many initializations is chosen. However, even these "bad" initializations very commonly give some partially-correct clustering behavior and may be combined to form a more accurate clustering algorithm. Our algorithm, which we refer to as Ensemble K-subspaces (EKSS), forms a co-association matrix whose (i, j)th entry is the number of times points i and j are clustered together by several runs of KSS with random initializations. We analyze the entries of this co-association matrix and show that a naïve version of our algorithm can recover subspaces under the same conditions as the Thresholded Subspace Clustering algorithm. We show on synthetic data that our method performs well even when the subspaces have large intersection or small principal angles, and when the data are noisy. Finally, we describe an "ensemble of ensembles" extension of our algorithm that achieves state-of-the-art performance across several benchmark datasets, including a resulting error for the COIL-20 database that is less than half that achieved by existing algorithms. This is joint work with David Hong and Dejiao Zhang.

1.2.4 Clustering Quality Measures for Subspace Clustering

In Chapter 5, we present ongoing work in the area of clustering quality evaluation for the specific case where the data lie on a union of subspaces. Since clustering is an inherently unsupervised problem, cross validation cannot be performed to evaluate the quality of the output from a given algorithm. While existing clustering quality measures (CQMs) such as the Dunn index [21] or Silhouette index [22] perform well on standard clustering datasets, they rely heavily on pairwise distances between points and are not applicable to UoS data. Further, subspace clustering algorithms such as Sparse Subspace Clustering [23] and Ensemble K-subspaces [24] require the selection of numerous parameters, and to the best of our knowledge there does not exist a principled means of choosing the appropriate parameters in the absence of ground-truth labels. In this chapter, we develop measures for subspace clustering quality that reliably predict clustering performance on

[1] The improvement over the runner-up, support vector machine (SVM) with Gaussian kernel, was not found to be statistically significant, i.e., these two algorithms are essentially tied. Further, the "no free lunch theorem" [20] indicates that all algorithms are equivalent when performance is averaged across all possible datasets. However, ensemble methods and SVMs remain popular choices with strong performance on many real-world problems.

both real and synthetic data drawn from a union of subspaces. We then develop a first take on an axiomatic study of subspace clustering quality similar to that of [25] and analyze our proposed CQMs in terms of these axioms.

1.2.5 Publications

Chapter 1

- J. Lipor, L. Balzano, B. Kerkez, and D. Scavia, "Quantile search: A distance-penalized active learning algorithm for spatial sampling," in *Allerton Conference on Communication, Control, and Computing*, 2015 [26].
- J. Lipor, B.P. Wong, D. Scavia, B. Kerkez, and L. Balzano, "Distance-penalized active learning using quantile search," accepted for publication in *IEEE Transactions on Signal Processing*, 2017 [27].

Chapter 2

- J. Lipor and L. Balzano, "Margin-based active subspace clustering," in *International Workshop on Computational Advances in Multi-Sensor Adaptive Processing* (CAMSAP), 2015 [28].
- J. Lipor and L. Balzano, "Leveraging union of subspace structure to improve constrained clustering," in *International Conference on Machine Learning* (ICML), 2017 [29].

Chapter 3

- J. Lipor, D. Hong, D. Zhang, and L. Balzano, "Subspace clustering using ensembles of K-subspaces," *arXiv preprint, arXiv: 1709.04744* [24].

Chapter 4

- J. Lipor and L. Balzano, "On Parameter Selection for Subspace Clustering," in preparation.

Other

- J. Lipor and L. Balzano, "Robust blind calibration via total least squares," in *International Conference on Acoustics, Speech, and Signal Processing* (ICASSP), 2014 [30].

1.3 Important Tools and Related Work

In this section, we discuss the recently-developed tools and results that are integral to the understanding of this thesis. The first set of results introduces the reader to the mindset of *active learning*, the setting of Chapters 2 and 3, where our goal is to quickly discover the most salient bits of information in a dataset. Next, we introduce a variety of tools from high-dimensional probability that extend classical results from mathematics and probability to a regime that is amenable to large-scale data analysis. These results are relied upon heavily in Chapters 3 and 4. The goal of introducing these results here is to build the intuition behind the major contributions of later chapters.

1.3.1 The Statistical Learning Setup

We begin this section by introducing the general statistical learning framework [31] and showing how the two models studied in this thesis fit as specific examples of this framework. Let $\mathcal{X}$ denote the set of data points or examples given to us, also known as the *input space* or *feature space*. When performing inference of any type (e.g., regression or classification), we let $\mathcal{Y}$ denote the set of possible inferred values. In the case of binary classification, we may set $\mathcal{Y} = \{0, 1\}$. We call the set of all functions that map from $\mathcal{X}$ to $\mathcal{Y}$ the *concept space* and typically consider a fixed subset of concepts known as the *hypothesis space* $\mathcal{H}$. We assume that there is an unknown joint distribution $\mathbb{P}_{XY}$ over $\mathcal{X} \times \mathcal{Y}$ from which we draw independent and identically distributed (i.i.d.) pairs (X_n, Y_n), $n = 1, \ldots, N$. Using these example-label pairs, our goal is to learn the concept $h \in \mathcal{H}$ that best predicts the label Y corresponding to the example X for *all* pairs (X, Y) drawn from P_{XY}. A general description of the means to learn such a hypothesis can be found in [31] and is beyond the scope of this document. Instead, we now demonstrate how each of the models considered in this thesis fit into the framework just described.

Example 1.1 (Hypoxia Sampling)**.** *In Chapter 2, we study the problem of determining the spatial extent of regions of low oxygen in Lake Erie using an autonomous watercraft. We model our feature space as the unit square intersected with the lake, i.e.,* $\mathcal{X} = [0, 1]^2 \cap \mathcal{L}$*, where* $\mathcal{L} \subset \mathbb{R}^2$ *is the set of points defined to be inside the lake. We classify a point* $x \in \mathcal{X}$ *as "hypoxic" if the oxygen concentration is below 2.0 ppm, in which case it has the corresponding label* $y = 0$ *(with the label* $y = 1$ *if the concentration is above this threshold). Hence* $\mathcal{Y} = \{0, 1\}$*. By breaking our two-dimensional estimation problem into a series of one-dimensional problems, we consider the simple hypothesis space of step functions,* $\mathcal{H} = \left\{h : [0, 1] \to \{0, 1\} \,|h(x) = \mathbf{1}_{[0,\theta)}(x)\right\}$*, where* $\theta \in [0, 1]$ *denotes the change point of the step function and* $\mathbf{1}_{\mathcal{S}}(x)$ *denotes the indicator function and takes the value 1 on the set* $\mathcal{S}$*. Thus, for each strip, our goal is to learn the change point* θ *that delineates the boundary between normal oxygen concentration and hypoxia.*

Example 1.2 (Nearest-Subspace Classifier)**.** *In Chapters 3 and 4, we study the problem of* subspace clustering*, in which our data points lie near one of K unknown low-dimensional linear subspaces, and we wish to classify them according to their nearest subspace. In Chapter 3, we leverage unsupervised estimates of these subspaces to inform our querying of example-label pairs.*[2] *In this case we have $\mathcal{X} = \{x_1, \dots, x_N\}$, a set of vectors we wish to classify, and $\mathcal{Y} = \{1, \dots, K\}$. Assuming we have a fixed set of K subspace bases $U_1, \dots, U_K$, we obtain our optimal concept immediately as*

$$h(x) = \arg\max_k \left\| U_k^T x \right\|.$$

Note that this formulation is non-standard and will not be used directly in subsequent chapters, but it will motivate the connection to other important tools presented in this chapter.

1.3.2 Tools from Active Learning

The first set of major results important to this thesis are those surrounding the topic of *active learning*. Suppose we are given the statistical learning setting above. In the classical learning setup, the labeled examples $x_1, \dots, x_N$ are fixed and chosen before the learning process begins. Contrasting this, in the active learning setting we are allowed to request labels for specific examples of our choice. Further, the example at time n can be chosen as a function of all previous examples $x_1, \dots, x_{n-1}$ and their labels $y_1, \dots, y_{n-1}$. The goal of active learning is to learn the best hypothesis $h^* \in \mathcal{H}$ using as few queries as possible, i.e., by minimizing N. In many cases, active learning algorithms have the benefit of achieving a given classification error by training on far fewer example-label pairs than their passive counterparts [6, 32, 33]. Two approaches to accomplishing this task dominate the literature, which we now describe.

The first approach to active learning is known as a "greedy" active learner [34] or "generalized binary search" [35, 36]. Consider the case where $\mathcal{H}$ is finite and contains a classifier consistent with all example-label pairs. Informally, the greedy approach states that we should choose each query such that the number of consistent classifiers is reduced by half. Consider, for example, the case of one-dimensional threshold classifiers defined in Example 1.1. Initially, the set of possible true thresholds is the entire unit interval. If the first sample is taken at $x_1 = 1/2$ with a label $y_1 = 1$, then we know immediately that the true threshold lies to the right of $1/2$ and half the possible thresholds have been removed from $\mathcal{H}$. This approach has been shown to discover the true classifier in an optimal number of queries, namely with $N = \mathcal{O}(\log |\mathcal{H}|)$ [34, 35]. However, a major drawback to this approach is that it is often impossible to determine which query achieves this "splitting" property, a fact that motivates the reduction to one-dimensional classifiers in Chapter 2.

[2] Technically, we query for *pairwise constraints* between points, but we consider the case of labels here to gain intuition.

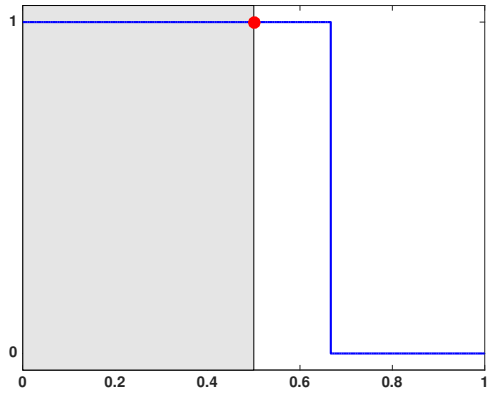

Figure 1.1: Example of greedy/binary search active learning algorithm. The remaining hypothesis space after the first measurement is [0.5,1].

The second approach to active learning is known as "uncertainty sampling" [37] or "opportunistic priors" [34]. The intuition behind this approach is to request labels for points that are most likely to be misclassified by our current concept. Suppose we wish to perform K-ary classification and have collected n example-label pairs, from which we train a classifier. For example, returning to Example 1.2 above, we may obtain K subspace bases $U_1, \cdots, U_K$. Assuming all points are normalized to have unit norm, we may treat the energy of a point in each basis as the posterior probability of assignment to that class under the current concept (see [37], Sec. 2.3), i.e.,

$$\mathbb{P}\left\{y = k|x\right\} = \left\|U_k^T x\right\|_2 .$$

Intuitively, if a point has roughly equal energy in many subspaces, our confidence in assigning that point is low, whereas if a point has unit energy in a single subspace and no energy in any other, our confidence is high. Many metrics for uncertainty exist, and we now discuss the two most common, known as *entropy* and *margin*. Given our model for the posterior probability of assignment, the point of maximum entropy is defined as [38]

$$x_H = \arg\max_{x \in \mathcal{X}} \; -\sum_{y=1}^{K} \mathbb{P}\left\{y|x\right\} \log \mathbb{P}\left\{y|x\right\} .$$

One major drawback to entropy sampling is that it favors points for which there is high overall uncertainty, as opposed to high uncertainty between two of the K classes. In the case where our goal is the minimization of the expected log-loss, entropy would be an appropriate choice [37]. However, if our goal is to minimize the number of misclassifications (0-1 loss), we must turn to another notion of uncertainty known as *margin*. Informally, a point is defined as having small margin if it lies near the decision boundary of the given classifier. Two notions of margin exist,

known as *additive margin* and *relative margin*, defined respectively as

$$x_{AM} = \arg\min_{x \in \mathcal{X}} \ \mathbb{P}\{y_1|x\} - \mathbb{P}\{y_2|x\}$$

and

$$x_{RM} = \arg\min_{x \in \mathcal{X}} \ 1 - \frac{\mathbb{P}\{y_2|x\}}{\mathbb{P}\{y_1|x\}},$$

where y_1 and y_2 denotes the most and second-most likely classifiers, respectively. Margin-based sampling has been shown to result in optimal rates of convergence in the case of linear support vector machines [33]. We extend the notion of relative margin to the case of unions of subspaces in Chapter 3 and show that an active learning algorithm can be used to improve subspace clustering algorithms using a small number of queries.

1.3.3 Tools from High-Dimensional Probability

The second set of major results important to this thesis can be found in the textbook [39]. While this book is excellent in its entirety, we explore a few key results that are especially important. We now state each of these results formally and describe briefly their impact on this work.

Much of statistical learning theory is based around bounding sums of independent random variables through concentration inequalities such as Hoeffding's inequality. These bounds typically become tighter as one increases the number of variables considered. Likewise, as the dimension of random variables increases, we can often bound the behavior of certain variables more tightly. One such class of variables that is often considered is that of *sub-Gaussian* random variables.

Definition 1.1 ([39], Prop. 2.5.2)**.** *We say that a random variable X with mean $\mu = \mathbb{E}[X]$ is sub-Gaussian if there is a positive number σ such that*

$$\mathbb{E}\left[e^{\lambda(X-\mu)}\right] \leq e^{\sigma^2\lambda^2/2}.$$

For a Gaussian random variable, the sub-Gaussian parameter σ is the standard deviation. Intuitively, a random variable is sub-Gaussian if its tails are Gaussian-like, and hence Hoeffding-type concentration inequalities exist to bound sums of variables of this class (see [39], Thm. 2.6.2). The first result in this section states that in high-dimensions, the norm of a sub-Gaussian random variable concentrates tightly around its mean.

Theorem 1.1 (Thm. 3.1.1 [39])**.** *Let $X \in \mathbb{R}^D$ be a random vector with independent, zero-mean, sub-Gaussian coordinates with parameter σ^2. Then*

$$\mathbb{P}\left\{\left|\|X\|_2 - \sqrt{D\sigma^2}\right| \geq t\right\} \leq 2e^{-ct^2/\max(\sigma^2,\sigma^4)},$$

where c is an absolute constant.

An interesting corollary from Thm. 1.1 is that in high dimensions, zero-mean Gaussian random vectors *concentrate tightly around the sphere in*, rather than around the origin as low-dimensional intuition might suggest. Letting $X \sim \mathcal{N}(0, \frac{1}{D} I_D)$, we see that

$$\mathbb{P}\left\{ \left| \|X\|_2 - 1 \right| \geq t \right\} \leq 2e^{-ct^2 D}.$$

This fact gives rise to the known similarity between spherical and Gaussian distributions, which is utilized in subspace clustering results such as [40].

Knowing what we can expect from the norm of a vector in high dimensions, the natural next question is to examine what happens to inner products between random vectors.

Lemma 1.1 (Lemma 3.2.4 [39])**.** *Let $X, Y \subset \mathbb{R}^D$ be isotropic, sub-Gaussian random vectors. Then*

$$\mathbb{E}\left[\left| \left\langle \frac{X}{\|X\|}, \frac{Y}{\|Y\|} \right\rangle \right| \right] = \frac{1}{\sqrt{D}}.$$

A second bit of intuition relies on the above lemma and tells us that *in high-dimensional space, all isotropic, sub-Gaussian vectors are nearly orthogonal*. This fact of high-dimensional geometry is leveraged in the Thresholded Subspace Clustering [41] algorithm referred to in Chapter 4. The basic premise behind this algorithm is that for two vectors in a d-dimensional subspace, the expected absolute value of their inner product is $1/\sqrt{d}$, compared to points in two random subspaces, which have expected inner product $1/\sqrt{D}$. Hence, we roughly expect points to be in the same subspace if their inner product is above a given threshold.

A final useful concentration inequality considers quadratic functions of random vectors. We first state the theorem and then describe its relation to the work in this thesis.

Theorem 1.2 (Hanson-Wright Inequality, Theorem 6.2.1 [39])**.** *Let $X \in \mathbb{R}^D$ be a random vector with independent, zero-mean, sub-Gaussian coordinates. Let $A \in \mathbb{R}^{D \times D}$. Then for every $t \geq 0$, we have*

$$\mathbb{P}\left\{ \left| X^T A X - \mathbb{E}\left[X^T A X \right] \right| \right\} \leq 2 \exp\left[-c \min\left(\frac{t^2}{\sigma^4 \|A\|_F^2}, \frac{t}{\sigma^2 \|A\|} \right) \right].$$

One question addressed in Chapter 4 of this thesis involves the projection of points in a subspace onto a *random* subspace basis. To gain intuition into how these projections should behave, we consider the following scenario. Let $X = \begin{bmatrix} X_1 & X_2 \end{bmatrix}$, where the columns of $X_1 \in \mathbb{R}^{D \times N_1}$ are distributed as $\mathcal{N}(0, \frac{1}{d} U_1 U_1^T)$ and $U_1 \in \mathbb{R}^{D \times d}$ is an orthonormal basis for a subspace. Let the N_2 columns of X_2 be similarly distributed in a subspace represented by $U_2 \in \mathbb{R}^{D \times d}$. We are interested in studying the Frobenius norm of the projection of X onto a random subspace basis U. By the

Hanson-Wright inequality, we have that

$$\left\|U^T X\right\|_F^2 \approx \frac{N_1}{d}\left\|U^T U_1\right\|_F^2 + \frac{N_2}{d}\left\|U^T U_2\right\|_F^2$$

with high probability. The above tells us that when we project onto a random subspace basis, we expect the amount of energy in the projection to be a function of (1) the number of vectors we project and (2) the similarity between the random subspace and the true subspaces. The term $\left\|U^T U_1\right\|_F^2$ is the square of the *subspace affinity* between the subspaces spanned by U and U_1, a notion of subspace closeness that we will describe further in the next section.

1.3.4 Subspace Distances

In Chapters 3 and 4, we consider the topic of subspace clustering, a problem whose difficulty increases as the distance between the underlying subspaces decreases. Just as centroid-based clustering algorithms like K-means fail as the cluster centers grow closer together, subspaces that are "nearby" are especially difficult to distinguish. To formalize this notion, we consider a few measures of distance between subspaces. The most common notion of distance between subspaces is found in [42] and deals with the *principal angles* between subspaces. First, let U_1 and U_2 be orthonormal bases for two d-dimensional subspaces in $\mathbb{R}^D$.

Definition 1.2 ([43], Defn. 2.6)**.** *The* principal angles $\theta_1 \leq \theta_2 \leq \cdots \leq \theta_d$ *between two subspaces* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *are recursively defined as*

$$\cos(\theta_k) = \max_{u \in \mathcal{S}_1, v \in \mathcal{S}_2} \frac{u^T v}{\|u\|_2 \|v\|_2} := \frac{u_k^T v_k}{\|u_k\|_2 \|v_k\|_2},$$

with the orthogonality constraints $u^T u_j = 0$, $v^T v_j = 0$, $j = 1, \ldots, k-1$. *Further, all principal angles may be found as*

$$\theta_k = \cos^{-1}(\sigma_k(U_1^T U_2)),\ k = 1, \ldots, d,$$

where $\sigma_k(Z)$ *denotes the* k*th largest singular value of the matrix* Z.

With the above definition, we now state a common definition of distance between subspaces, which is equal to the sine of the smallest principal angle between the two subspaces.

Definition 1.3 ([42], Sec. 2.5.3)**.** *Consider two subspaces* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *with corresponding orthonormal bases* U_1 *and* U_2 *and orthogonal projection matrices* P_1 *and* P_2. *The* distance *between two*

subspaces is defined as

$$\begin{aligned} \operatorname{dist}(\mathcal{S}_1, \mathcal{S}_2) &= \|P_1 - P_2\|_2 \\ &= \sigma_{\max}(P_1 - P_2) \\ &= \left\|U_1^T U_2^\perp\right\|_2 = \left\|U_2^T U_1^\perp\right\|_2 \\ &= \sin(\theta_{\max}), \end{aligned}$$

where $U^\perp = I - UU^T$ denotes the projection onto the orthogonal complement of the space spanned by U, and $\theta_{\max}$ is the maximum principal angle between the subspaces.

Note that the subspace distance is between 0 and 1 and is zero if and only if the two subspaces are the same. A distance of 1 implies that the *maximum* angle between subspaces is $\pi/2$, but it does not tell us about the other $d-1$ principal angles. For this reason, another measure of distance between subspaces known as the *subspace affinity* is commonly considered.

Definition 1.4 ([43], Defn. 2.7)**.** *The* affinity *between two subspaces is*

$$\begin{aligned} \operatorname{aff}(\mathcal{S}_1, \mathcal{S}_2) &= \sqrt{\frac{1}{d}\sum_{k=1}^{d} \cos^2 \theta_k} \\ &= \frac{1}{\sqrt{d}} \left\|U_1^T U_2\right\|_F . \end{aligned}$$

The affinity defined above is between 0 and 1, with a value of 0 implying the subspaces are orthogonal in all directions. The subspace affinity is stronger in the sense that it captures information about all angles between subspaces, rather than the maximum only.

A commonly-analyzed model in subspace clustering is that of the *fully random* model, in which K subspaces are drawn uniformly at random from the set of all subspaces, and the points within each subspace are drawn uniformly at random from the unit sphere intersected with their respective subspace. In such a setting, a natural quantity to identify is the expected affinity between two subspaces. We now give an informal reasoning based on the tools introduced above. The argument can be made rigorous via [40], Lemma B.4a. In sufficiently high ambient dimension, Lemma 1.1 tells us that random unit-norm vectors are approximately orthogonal. Thus, we model a random subspace basis as a set of d vectors drawn uniformly at random from S^{D-1}, where D is the ambient

dimension. Let U and V denote two such orthonormal bases. Then we have

$$\begin{aligned}
\mathbb{E}\left\|U^TV\right\|_F^2 &= \mathbb{E}_V\mathbb{E}_{U|V}\left\|U^TV\right\|_F^2 \\
&= \mathbb{E}_V\mathbb{E}_{U|V}\mathrm{tr}\left(V^TUU^TV\right) \\
&= \mathrm{tr}\left(\mathbb{E}_VV^T\left[\mathbb{E}_{U|V}UU^T\right]V\right) \\
&= \mathrm{tr}\left(\mathbb{E}_VV^T\frac{d}{D}I_DV\right) \\
&= \frac{d}{D}\mathrm{tr}\left(\mathbb{E}_VV^TV\right) \\
&= \frac{d^2}{D},
\end{aligned}$$

where we have used the fact that for $u_i \sim \text{Unif}\left(S^{D-1}\right)$,

$$\mathbb{E}u_iu_i^T = \frac{1}{D}I_D.$$

This result gives us a bit of intuition about the subspace clustering problem. In the fully random model, the problem becomes easier as the ambient dimension grows and more difficult as the subspace dimension grows.

1.4 Summary

In this thesis, we tackle the challenges of modern data analysis by utilizing key features inherent to large datasets. By seeking out the important examples in massive datasets and leveraging underlying low-dimensional structure, we are able to develop and analyze algorithms capable of translating the overwhelming influx of data into useful information.

CHAPTER 2

Quantile Search: An Active Learning Algorithm for Spatial Sampling

2.1 Introduction

This chapter includes collaborative work as part of the M-Cubed program with Brandon Wong and Branko Kerkez in the department of Civil and Environmental Engineering and Donald Scavia in the School of Natural Resources and Environment. The work of the experimental section (Sec. 2.4.4) was primarily performed by Brandon Wong and others in the Real-Time Water Systems Lab. Finally, we would like to thank Hye Won Chung for her insights into the information-theoretic interpretation of probabilistic binary search.

Intelligently sampling signals of interest has been a fundamental topic in the signal processing community for many years, the most recent advances in this area being compressed sensing [44] and active learning [37]. In these and other scenarios, the goal is typically to recover a signal from a given class (e.g., bandlimited signals or the Bayes decision boundary for $0/1$ signals) using as few samples as possible. However, in the modeling of spatial phenomena, such as oxygen concentration in lakes, the sampling cost is a function of both the number of samples required *and* the cost to travel to the sample locations. Therefore, the design of provably efficient algorithms to detect spatial phenomena is an important open problem and is the topic of this chapter.

Consider our motivating problem, in which we wish to estimate the boundary of a hypoxic region (i.e., a region of oxygen concentration below 2.0 ppm [3]) in the central basin of Lake Erie using an autonomous watercraft with a speed ranging from 0.5-4 m/s. Fig. 1 shows an interpolated estimate of the oxygen concentration based on a small number of samples taken throughout the lake, where the hypoxic zone is denoted by the dark region (in color: blue/purple region). Oxygen concentration is a strong indicator of the health of the Great Lakes [3] and the spatial extent of such regions is a topic of interest for researchers in the field [4, 45]. We assume the hypoxic region is connected with a smooth boundary and that the boundary remains relatively stationary over the course of a few days. The problem of estimating the boundary can then be viewed as a binary

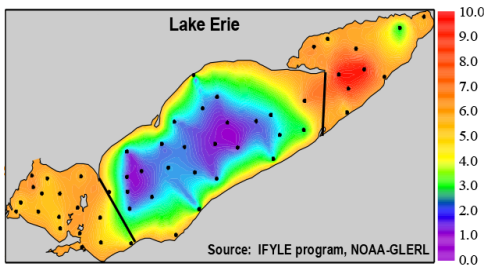

Figure 2.1: Dissolved oxygen concentrations in Lake Erie. Points represent sample locations and solid black lines delineate the central basin.

classification problem, in which spatial points receive a label 0 if they are hypoxic and 1 otherwise, and the desired spatial extent corresponds to the Bayes decision boundary. Our goal is to learn the decision boundary in as little *time* as possible.

While the application of optimal active learning algorithms such as [5, 46] minimizes the number of samples required to estimate the boundary, little attention has been given to additional penalties that affect the cost of sampling. In the case of sampling in Lake Erie, the distance traveled between all sampling locations is on the order of hundreds of kilometers, and thus algorithms such as [5, 46], which require a coarse sampling of the entire feature space, are not applicable.

In this chapter, we present an active learning algorithm called *quantile search* that achieves a tradeoff between the number of measurements and distance traveled to estimate the change point of a one-dimensional step function. At its two extremes, quantile search minimizes either the number of samples or the distance traveled to estimate the decision boundary, with a tradeoff achieved by varying a search parameter. We derive the expected number of samples required and distance traveled in the noiseless case and bound the number of samples required in the case of noisy measurements. We also show how a series of one-dimensional estimates can be used to estimate the two-dimensional boundary of interest. We present a novel generalization in the case of noisy measurements that, unlike our initial work [26], is equivalent to the noiseless case when the probability of measurement error is zero. We also provide two algorithmic improvements for the problem of interest and show in simulations that these greatly reduce the required sampling time. Our simulations are realistic, including real bathymetry data from Lake Erie provided by the National Oceanic and Atmospheric Administration [47]. We also compare the performance of our algorithm to a version of *proactive learning* [2]. Finally, we include results of our experiments performed on Third Sister Lake in Ann Arbor, MI with an autonomous watercraft controlled using a cloud-based architecture.

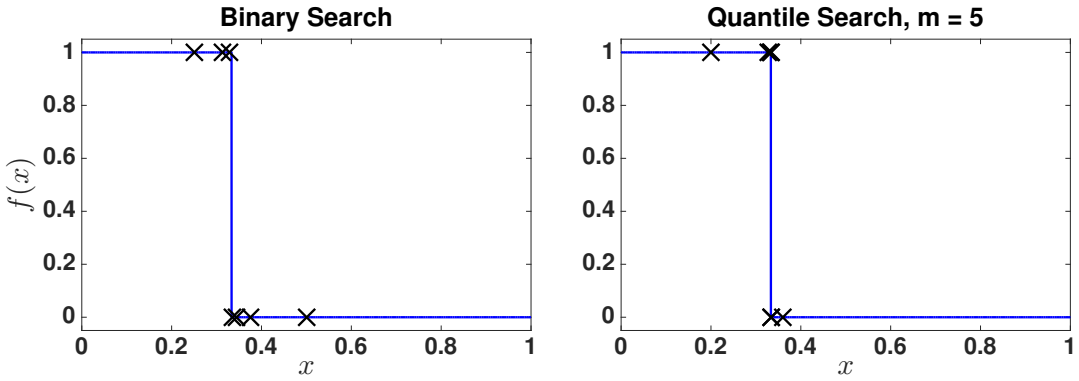

Figure 2.2: Example step function with $\theta = 1/3$ with corresponding measurements (marked by an x) taken using binary search (left) and quantile search with $m = 5$ (right).

2.2 Problem Formulation & Related Work

Determining the spatial extent of the hypoxic region shown in Fig. 2.1 can be interpreted as learning a two-dimensional Bayes decision boundary. Following [6], we split our two-dimensional problem into several one-dimensional intervals, a process that is described further in Section 2.4 and can be viewed in Fig. 2.7a-2.7d. The idea here is that we can carve a two-dimensional boundary fragment (indeed any d-dimensional boundary fragment class) into several one-dimensional interval problems, piecing the solutions together for a full boundary estimate.[1]

Having reduced the problem to several one-dimensional problems, on each interval we must find a threshold beyond which the lake is hypoxic. Define the step function class

$$\mathcal{F} = \{f : [0,1] \to \mathbb{R} | f(x) = \mathbf{1}_{[0,\theta)}(x)\}$$

where $\theta \in [0,1]$ is the change point and $\mathbf{1}_S(x)$ denotes the indicator function, which is 1 on the set S and 0 elsewhere. An example function belonging to $\mathcal{F}$ with $\theta = 1/3$ is shown in Fig. 2.2. In contrast to the standard active learning scenario, our goal is to estimate θ while minimizing the total time required for sampling, a function of both the number of samples taken *and* the distance traveled. Denote the observations $\{Y_n\}_{n=1}^N \in \{0,1\}^N$ as samples of an unknown function $f_\theta \in \mathcal{F}$ taken at sample locations on the unit interval $\{X_n\}_{n=1}^N$. With probability p, $0 \leq p < 1/2$, we observe an erroneous measurement. Thus

$$Y_n = \begin{cases} f_\theta(X_n) & \text{with probability } 1-p \\ 1 - f_\theta(X_n) & \text{with probability } p \end{cases} = f(X_n) \oplus U_n,$$

[1] As we discuss in Section 2.2.1, this is order-optimal in terms of sample complexity. Our heuristic algorithmic improvements of Section 2.3.3 allow us to more intelligently sample from one interval to the next.

where $\oplus$ denotes summation modulo 2, and $U_n \in \{0, 1\}$ are Bernoulli random variables with parameter p. While other noise scenarios are common, here we assume the U_n are independent and identically distributed and independent of $\{X_n\}$. This noise scenario is of interest as the motivating data (oxygen concentration) is a thresholded value in $\{0, 1\}$, where Gaussian noise results in improper thresholding of the measurements. The extension to nonuniform noise (e.g., a Tsybakov-like noise condition as studied in [46]) remains as a topic for future work.

2.2.1 Related Work

A number of active learning algorithms designed to estimate θ exist; however, these algorithms typically assume the sampling cost is due only to the measurements themselves. Most similar to our algorithm is the method of binary bisection and its extensions [48, 49, 50, 51, 6, 46, 52]. In the noiseless case, binary bisection estimates the change point of a step function on the unit interval by successively halving the space of potential classifiers, termed the *hypothesis space*. An example of this search procedure is shown in the left-hand plot of Fig. 2.2. A noise-tolerant version of this algorithm was first presented in [48], where measurements are flipped with known probability p. A discretized version of this algorithm was analyzed in [49] and shown to be minimax optimal in [46] under the Tsybakov noise condition. Further, the authors of [46] use the discretized algorithm to show that a series of one-dimensional threshold estimates can be used to estimate functions belonging to the boundary fragment class in d dimensions at a minimax optimal rate. The original algorithm presented in [48] was recently shown to converge at a geometric rate in [52]. Binary bisection has also been used to obtain optimal rates in optimization [53] and in the noisy 20 questions problem [54]. In [6], the authors give a spatial sampling problem as motivation for the probabilistic binary search (PBS) algorithm. However, a simple analysis shows that in the noiseless case, to estimate the threshold of a step function on the unit interval, binary search travels the entire unit interval. Hence, while the worst-case number of samples required is minimized, the total distance traveled is the worst possible. In the motivating problem given above, the central basin of Lake Erie has a width of roughly 80 km, making this approach prohibitive.

More sophisticated active learning algorithms have been widely studied, achieving optimal rates for piecewise constant functions in [46] and for the linear support vector machine in [33]. In both cases, the algorithm begins by uniformly sampling the entire feature space. Again considering the problem of interest, the central basin of Lake Erie has an area of approximately 14,000 km^2, making this approach infeasible. In contrast, the algorithm studied in [46] was used in [5] to measure the hydrodynamics of Lake Wingra in Madison, WI, which has an area of 1.3 km^2.

Nonuniform sampling costs are studied in [55, 56, 57, 2]. In [55], the authors use the uncertainty sampling heuristic to determine the most informative points and penalize for spatial costs using the

traveling salesman problem with profits. The work of [56] uses both uncertainty and diversity to select points and also penalizes for arbitrary costs. In both cases, the algorithm proceeds in batches, i.e., by iteratively requesting a set of labels and retraining the classifier. This approach suffers the same pitfalls as [5] in that the algorithm can require traversing the entire feature space multiple times. Further, neither algorithm is accompanied by theoretical guarantees. In [57], the authors present and analyze a greedy algorithm for active learning with nonuniform costs. However, in our case the cost associated with each point is the distance from the previous point, so the costs in question are both nonuniform and *dynamic*. A somewhat similar algorithm, known as *proactive learning*, is presented in [2], where the proposed strategy chooses at each round the point maximizing the difference between or ratio of informativeness and cost to label the point. In Section 2.4, we compare with this algorithm using mutual information as our metric for informativeness.

The problem of sampling spatial phenomena using mobile robots has been studied in signal processing and robotics literature as well. In [58, 59], the authors study the case where the sampling cost is near-zero and show that equispaced parallel lines result in the minimum distance required to reconstruct a variety of practical signals. Mutual information is also used as a metric for informativeness in [60], where the authors impose a Gaussian process model to perform path planning for robots used to track a variety of spatial phenomena. A greedy algorithm is presented with theoretical guarantees based on submodularity [61]. However, the model imposed is not appropriate for determining the boundary of a region of interest. The recent work of [62] considers the measurement cost and travel time to estimate the location of point targets using mobile robots but does not easily extend to the case of estimating the boundary of a region of interest. The algorithm in [63] is similar to the one described in [2], with the main difference being that in early stages the algorithm emphasizes regularity of samples (i.e., encourages early samples to be taken uniformly throughout the feature space).

2.3 Quantile Search

In this section, we present our algorithm *quantile search*, an extension of binary search and ideas in [49, 6] to penalize both the sample complexity and distance traveled during the estimation procedure. The basic idea behind this algorithm is as follows. We wish to find a tradeoff between the number of samples required and the total distance traveled to achieve a given estimation error for the change point of a step function on the unit interval. As we know, binary bisection minimizes the number of required samples. On the other hand, continuous spatial sampling minimizes the required distance to estimate the threshold. Binary search bisects the feasible interval (hypothesis space) at each step. In contrast, one can think of continuous sampling as dividing the feasible interval into infinitesimal subintervals at each step. With this in mind, a tradeoff becomes clear: one can divide the feasible

Algorithm 2.1 Deterministic Quantile Search (DQS)

Input: search parameter m, stopping error ε
Initialize: $X_0 \leftarrow 0$, $Y_0 \leftarrow 1$, $n \leftarrow 1$, $a \leftarrow 0$, $b \leftarrow 1$
while $b - a > 2\varepsilon$ **do**
  if $Y_{n-1} = 1$ **then**
    $X_n \leftarrow X_{n-1} + \frac{1}{m}(b - a)$
  else
    $X_n \leftarrow X_{n-1} - \frac{1}{m}(b - a)$
  end if
  $Y_n \leftarrow f(X_n)$
  $a = \max\{X_i : Y_i = 1, i \leq n\}$
  $b = \min\{X_i : Y_i = 0, i \leq n\}$
  $\hat{\theta}_n \leftarrow \frac{a+b}{2}$
end while

interval into subintervals of size $1/m$, where m is a real number between 2 and ∞. Intuition would tell us that increasing m would increase the number of samples required but decrease the distance traveled in sampling. In what follows, we show that this intuition is correct in both the noise-free and noisy cases, resulting in two novel search algorithms.

2.3.1 Deterministic Quantile Search

We first describe and analyze quantile search in the noise-free case ($p = 0$), here referred to as deterministic quantile search (DQS). To estimate the change point of a step function, deterministic binary bisection travels either forward or backward (depending on the measurement) a fraction 1/2 into the feasible interval. In contrast, the DQS algorithm presented here travels $1/m$ forward or backward, where $m \in [2, \infty)$. While the DQS measurements for $m > 2$ are less informative than in binary bisection, we expect that the distance traveled during the estimation procedure will be reduced, since we can pass the change point by a fraction at most $1/m$. The search procedure for the case of $m = 5$ is shown in the right-hand plot of Fig. 2.2. Note that in contrast to binary search, quantile search does not overshoot the change point $\theta = 1/3$ by a significant amount. A formal description of the procedure is given in Algorithm 2.1. In the following subsections, we analyze the expected sample complexity and distance traveled for the algorithm and show the required number of samples increases monotonically with m, and the distance traveled decreases monotonically with m, indicating that the desired tradeoff is achieved.

2.3.1.1 Convergence of Estimation Error

We analyze the expected error after a fixed number of samples for the DQS algorithm. The main result and a sketch of the proof are provided here. An expanded proof can be found in Appendix A.

Theorem 2.1. *Consider a deterministic quantile search with parameter m and let $\rho = \frac{m-1}{m}$. Begin with a uniform prior on θ. The expected estimation error after n measurements is then*

$$\mathbb{E}[|\hat{\theta}_n - \theta|] = \frac{1}{4}\left[\rho^2 + (1-\rho)^2\right]^n. \tag{2.1}$$

Proof. (Sketch; see complete proof in Appendix A) The proof proceeds from the law of total expectation. Let $Z_n = |\hat{\theta}_n - \theta|$. The first measurement is taken at $1/m$, and hence the expected error can be calculated when $\theta \leq 1/m$ and $\theta > 1/m$.

$$\begin{aligned}\mathbb{E}[Z_1] &= \mathbb{E}\left[Z_1 | \theta \leq \frac{1}{m}\right]\mathbb{P}\left(\theta \leq \frac{1}{m}\right) + \\ &\quad \mathbb{E}\left[Z_1 | \theta > \frac{1}{m}\right]\mathbb{P}\left(\theta > \frac{1}{m}\right) \\ &= \frac{1}{4}\left[(1-\rho)^2 + \rho^2\right].\end{aligned}$$

Similarly, after the second measurement is taken, there are four intervals, two which partition the interval $[0, 1/m]$, and two which partition $(1/m, 1]$. These result in four monomials of degree 4, one of which is $(1-\rho)^4$, one which is ρ^4, and two which are $(1-\rho)^2\rho^2$. The basic idea is that each "parent" interval integrates to $(1-\rho)^i\rho^j$ and in the next step gives birth to two "child" intervals, one evaluating to $(1-\rho)^{i+1}\rho^j$ and the other $(1-\rho)^i\rho^{j+1}$. The proof of the theorem then follows by induction. □

Consider the above result when $m = 2$. In this case, the error becomes $\mathbb{E}[|\hat{\theta}_n - \theta|] = 2^{-(n+2)}$. Comparing to the worst case, we see that the average case sample complexity is exactly one sample better than the worst case, matching the well-known theory of binary search. In Section 2.4 we confirm this result through simulation.

2.3.1.2 Distance Traveled

Next, we analyze the expected distance traveled by the DQS algorithm in order to converge to the true θ. The proof is similar to that of the previous theorem in that it follows by the law of total expectation. After each sample, we analyze the expected distance given that the true θ lies in a given interval. The result and a proof sketch are given below, with the full proof included in Appendix A.

Theorem 2.2. *Let D denote a random variable representing the distance traveled during a deterministic quantile search with parameter m. Begin with a uniform prior on θ. Then*

$$\mathbb{E}[D] = \frac{m}{2m-2}. \tag{2.2}$$

Proof. (Sketch, see full proof in Appendix A) We first consider the expected distance traveled before the algorithm reaches a point $x_1 > \theta$. Let D_1 be a random variable denoting this distance. Once the algorithm passes this point, it moves in the reverse direction until reaching $x_2 < \theta$, moving a distance D_2. This process repeats until convergence. Let D_n be a random variable denoting the distance required to move to the right of θ for the $\lceil \frac{n}{2} \rceil$th time when n is odd, and to the left of θ for the $\frac{n}{2}$th time when n is even. In this case, we have that

$$\mathbb{E}[D] = \sum_{n=1}^{\infty} \mathbb{E}[D_n]. \tag{2.3}$$

First, we would like to find $\mathbb{E}[D_1]$. Let A_i denote the interval $\left[\frac{1}{m}\sum_{p=0}^{i-1}\left(\frac{m-1}{m}\right)^p, \frac{1}{m}\sum_{p=0}^{i}\left(\frac{m-1}{m}\right)^p\right)$, where $A_0 = \left[0, \frac{1}{m}\right)$, so that the A_i's form a partition of the unit interval whose endpoints are possible values of the sample locations X_j. Now note that

$$\mathbb{E}[D_1] = \sum_{i=0}^{\infty} \mathbb{E}[D_1 | \theta \in A_i]\mathbb{P}(\theta \in A_i).$$

Then since we assume θ is distributed uniformly over the unit interval,

$$\begin{aligned}
\mathbb{P}(\theta \in A_i) &= \frac{1}{m}\sum_{p=0}^{i}\left(\frac{m-1}{m}\right)^p - \frac{1}{m}\sum_{p=0}^{i-1}\left(\frac{m-1}{m}\right)^p \\
&= \frac{1}{m}\left(\frac{m-1}{m}\right)^i.
\end{aligned}$$

Next, note that

$$\begin{aligned}
\mathbb{E}[D_1 | \theta \in A_i] &= \frac{1}{m}\sum_{p=0}^{i}\left(\frac{m-1}{m}\right)^p \\
&= 1 - \left(\frac{m-1}{m}\right)^{i+1}.
\end{aligned}$$

Thus we have

$$\begin{aligned}\mathbb{E}[D_1] &= \sum_{i=0}^{\infty} \mathbb{E}[D_1|\theta \in A_i]\mathbb{P}(\theta \in A_i) \\ &= \sum_{i=0}^{\infty} \left[1 - \left(\frac{m-1}{m}\right)^{i+1}\right]\left[\frac{1}{m}\left(\frac{m-1}{m}\right)^{i}\right] \\ &= \frac{m}{2m-1}.\end{aligned}$$

The proof proceeds by rewriting the above in terms of $\rho = (m-1)/m$ and then calculating $\mathbb{E}[D_n]$. This is done by dividing each A_i into subintervals which form partitions of A_i. By induction we get

$$\mathbb{E}[D_n] = \frac{m}{(2m-1)^n}\ , \tag{2.4}$$

and the result then follows from the infinite sum of (2.3). □

2.3.1.3 Sampling Time

Using the above results, we wish to find the optimal tradeoff for a given set of sampling parameters. Let γ be the time required to take one sample and η be the time required to travel one unit of distance. The total sampling time T is then

$$T = \gamma N + \eta D, \tag{2.5}$$

where N denotes the number of samples required. Given a fixed sampling time and desired error, (2.5) can be used to estimate the sample budget N. However, this approach differs from our goal of minimizing the total sampling time. Alternatively, the average value of N can be estimated numerically and used to optimize the expected value of T. We show examples of this approach in Section 2.4 for both the deterministic and probabilistic versions of quantile search.

As a final note, one may wonder about the relation to what is known as m-ary search [64]. In contrast to quantile search, m-ary search is tree-based. To make the difference clear, consider an example with $\theta = 3/8$ and let $m = 4$. In this case, both algorithms take their first sample at $X = 1/4$. However, after measuring $Y = 1$, quantile search takes its second measurement at $X = 7/16$, while m-ary search proceeds to $X = 1/2$. One may then expect that both algorithms would achieve the desired tradeoff, with m-ary search using fewer samples and more distance for the same value of m. We focus on quantile search for two reasons. First, quantile search does not require m to be an integer and therefore gives more flexibility in the resulting tradeoff. Second, quantile search as described is the natural generalization of PBS and lends itself to the analysis of

[49, 6] in the case where the measurements are noisy. A comparison to noisy m-ary search is a topic for future work.

2.3.2 Probabilistic Quantile Search

In this section, we extend the idea behind Section 2.3.1 to the case where measurements may be noisy (i.e., $p \geq 0$). In [49], the authors present an algorithm referred to in the literature as *probabilistic binary search* (PBS). The basic idea behind this algorithm is to perform Bayesian updating in order to maintain a posterior distribution on θ given the measurements and locations. Rather than bisecting the interval at each step, the algorithm bisects the posterior distribution. This process is then iterated until convergence and has been shown to achieve optimal sample complexity throughout the literature [46, 51]. We now extend this idea using the quantile methodology of the previous section, resulting in what we term *probabilistic quantile search* (PQS).

The idea behind PQS is straightforward. Starting with a uniform prior, the first sample is taken at $X_1 = 1/m$. The posterior density $\pi_n(x)$ is then updated as described below, and $\hat{\theta}_n$ is chosen as the median of this distribution. The algorithm proceeds by taking samples X_n such that

$$\int_0^{X_{n+1}} \pi_n(x)dx = \frac{1}{m}.$$

For $m = 2$, the above denotes the median of the posterior distribution and reduces to PBS, while in general this denotes sampling at the m-quantile of the posterior. A formal description is given in Algorithm 2.2.

We derived the update for PQS in our previous work [26], and it can be seen in steps 7 and 9 of Algorithm 2.2. Here we derive a more general version of the update that will be referred to in Section 2.4.2. Begin with the first sample. We have $\pi_0(x) = 1$ for all x and wish to find $\pi_1(x)$. Let $f_1(x|X_1, Y_1)$ be the conditional density of θ given X_1, Y_1. Applying Bayes rule, the posterior becomes:

$$f_1(x|X_1, Y_1) = \frac{\mathbb{P}(X_1, Y_1|\theta = x)\pi_0(x)}{\mathbb{P}(X_1, Y_1)}$$

For illustration, consider the case where $\theta = 0$. We now take the first measurement at $X_1 = \phi$ (note $\phi = 1/m$ for PQS). Then

$$\mathbb{P}\left(X_1 = \phi, Y_1 = 0|\theta = 0\right) = 1 - p$$

and

$$\mathbb{P}\left(X_1 = \phi, Y_1 = 1|\theta = 0\right) = p.$$

Algorithm 2.2 Probabilistic Quantile Search (PQS)

Input: search parameter m, probability of error p
Initialize: $\pi_0(x) = 1$ for all $x \in [0,1]$, $n \leftarrow 0$
while not converged **do**
  choose X_{n+1} such that $\int_0^{X_{n+1}} \pi_n(x)dx = \frac{1}{m}$
  $Y_{n+1} \leftarrow f(X_{n+1}) \oplus U_{n+1}$, where $U_{n+1} \sim$ Ber(p)
  if $Y_{n+1} = 0$ **then**

$$\pi_{n+1}(x) = \begin{cases} (1-p)\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x \leq X_{n+1} \\ p\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x > X_{n+1} \end{cases}$$

  else

$$\pi_{n+1}(x) = \begin{cases} p\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x \leq X_{n+1} \\ (1-p)\left(\frac{m}{1+(m-2)p}\right)\pi_n(x), & x > X_{n+1} \end{cases}$$

  end if
  $n \leftarrow n+1$
end while
estimate $\hat{\theta}_n$ such that $\int_0^{\hat{\theta}_n} \pi_n(x) = 1/2$

In fact, this holds for any $\theta < \phi$. Now examine the denominator:

$$\begin{aligned} \mathbb{P}(X_1 = \phi, Y_1 = 0) &= \phi(1-p) + (1-\phi)p \\ &:= \phi * p, \end{aligned}$$

We then update the posterior distribution to be

$$\pi_1(x) = \begin{cases} \frac{(1-p)}{\phi * p} & x \leq \phi \\ \frac{p}{\phi * p} & x > \phi. \end{cases}$$

The equivalent posterior density can be found for when $Y_1 = 1$. The process of making an observation and updating the prior is then repeated, yielding general formula for the posterior update. When $Y_{n+1} = 0$, we have

$$\pi_{n+1}(x) = \begin{cases} \frac{(1-p)}{\phi * p}\pi_n(x) & x \leq X_{n+1} \\ \frac{p}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

Similarly, for $Y_{n+1} = 1$, we have

$$\pi_{n+1}(x) = \begin{cases} \frac{p}{\phi * p}\pi_n(x) & x \leq X_{n+1} \\ \frac{(1-p)}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

2.3.2.1 Convergence of Estimation Error

Analysis of the above algorithm has proven difficult since its inception in 1974, with a first proof of a geometric rate of convergence appearing only recently in [52]. Instead, the authors and those following use a discretized version involving minor modifications. We follow this strategy, with the discretized algorithm given in Appendix A. In this case, the unit interval is divided into bins of size Δ, such that $\Delta^{-1} \in \mathbb{N}$. The posterior distribution is parameterized, and a parameter α is used instead of p in the Bayesian update, where $0 < p < \alpha$. The analysis of rate of convergence then centers around the increasing probability that at least half of the mass of $\pi_n(x)$ lies in the correct bin. A formal description of the algorithm can be found in Appendix A. Given this discretized version of PQS, we arrive at the following result.

Theorem 2.3. *Under the assumptions given in Section 2.2, the discretized PQS algorithm satisfies*

$$\sup_{\theta \in [0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq 2\left(\frac{m-1}{m} + \frac{2\sqrt{p(1-p)}}{m}\right)^{n/2}. \tag{2.6}$$

The proof can be found in Appendix A. In the case where $m = 2$, the above result matches that of [49, 6] as desired. One important fact to note is that in contrast to the deterministic case, the result here is an upper bound on the number of samples required for convergence as opposed to an expected value. As this seems to be the case for all analyses of similar algorithms [49, 6, 52], we instead rely on Monte Carlo simulations to choose the optimal value of m. Finally, the bound here is loose. For clarity, consider the case where $p = 0$ and $m = 2$. Then the above becomes

$$\sup_{\theta \in [0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq 2\left(\frac{1}{2}\right)^{n/2}.$$

As noted in [46], we can see by inspection that

$$\sup_{\theta \in [0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq \left(\frac{1}{2}\right)^{n+1},$$

indicating that we lose a factor of about $n/2$, even for the PBS algorithm bound in [46]. However, in [46], the authors use this result when $m = 2$ to show rate optimality of the PBS algorithm. This

fact suggests that despite the discrepancy, the result of Thm 2.3 may still be useful in proving some sort of optimality for the PQS algorithm.

While the rate of convergence for PQS can be derived using standard techniques, the expected distance or a useful bound on the distance is more difficult. The technique used in Section 2.3.1 becomes intractable as the values of X_n are no longer deterministic. The approach of examining the posterior distribution after each step and calculating the possible locations has been examined, but at the nth measurement, there are 2^{n-1} possible distributions. Further, PQS as described above has the undesirable property that it does not always travel toward the median of the distribution—a problem we overcome in the next section—and hence the distance traveled is higher than strictly necessary, making analysis of its distance properties of minimal practical importance.

2.3.2.2 Truncated PQS

Probabilistic quantile search as presented in Algorithm 2.2 is not a strict generalization of DQS in the sense that the two algorithms are not equivalent in the noiseless case. Moreover, in some cases, PQS will choose a sample location farther away from the current location than the median. This choice is suboptimal, as the median of the posterior is the most informative point (in an information-theoretic sense), and hence traveling farther to obtain less information is contrary to our overall goal. For these reasons, we propose the following variant of PQS, which has a sample complexity and distance traveled no worse than the PQS algorithm in Algorithm 2.2. The algorithm satisfies the statement of Thm. 2.3 (see Appendix A), and we show the improved performance in terms of both distance and sample complexity in Section 2.4. Instead of taking a sample at the m-quantile of the posterior, we instead truncate the posterior distribution in such a way to maintain the median as well as guarantee that the m-quantile of this truncated posterior is moving our sampling location towards the median of the posterior (the most informative point). We refer to this algorithm as Truncated PQS (TPQS).

Truncated PQS begins by sampling at the m-quantile as in PQS. For subsequent samples, we first define

$$\chi = \min\left\{\int_0^{X_n} \pi_n(x)dx, \int_{X_n}^1 \pi_n(x)dx\right\},$$

the probability in the tail of the distribution that would possibly cause us to move away from the median point. We then define the truncated distribution to be the normalized form of

$$\tilde{\pi}_n(x) = \begin{cases} 0, & \int_0^x \pi_n(z)dz \leq \chi \\ 0, & \int_x^1 \pi_n(z)dz \leq \chi \\ \pi_n(x), & \text{otherwise} \end{cases}.$$

Finally, the sample location is chosen as

$$X_{n+1} = \underset{X \in \left\{\tilde{X}_0, \tilde{X}_1\right\}}{\arg\min} |X_n - X| \ ,$$

where

$$\int_0^{\tilde{X}_0} \tilde{\pi}(x)dx = \frac{1}{m} \quad \text{and} \quad \int_{\tilde{X}_1}^1 \tilde{\pi}(x)dx = \frac{m-1}{m}.$$

Analogous to traveling "forward" or "backward" in DQS, this process guarantees that we always choose sample locations that are in the direction of the median of the posterior. This fact ensures that the information gain is at least that of the PQS algorithm, while choosing the nearer of the two locations results in a distance no greater than that of PQS. Note that we continue to use $\pi_n(x)$ as the posterior distribution of θ and update this distribution according to Algorithm 2.2, i.e., we only use $\tilde{\pi}_n(x)$ when choosing the sample locations. Further, for the case of $m = 2$, this generalization and PQS are equivalent, both resulting in the PBS algorithm.

2.3.2.3 Stopping Criterion

Previous work on PBS centers around the case where there is a fixed sample budget, avoiding the need for a stopping criterion for this algorithm. However in our application, while we need to further reduce sampling resources, we only stop sampling once we have reached a desired accuracy. In this case, one natural choice of stopping criteria for PBS would be to stop when the distance between successive samples is smaller than some predetermined value. However, in the case of PQS with high m, the step size may be very small from the start, resulting in early termination. In the case of DQS, the width of the feasible interval provides a direct measure of the absolute error in estimating θ. While there is no such width in the case of PQS, the certainty in our estimate of θ is quantified via the posterior distribution $\pi_n(x)$, which is discretized in our implementation. In light of this, we terminate PQS (or its generalized version) when there exists an x_i such that $\pi_n(x_i) \geq 0.9$.

2.3.3 Algorithmic Improvements

In this section, we describe two heuristics that can be used to further reduce the sampling time. These heuristics are appropriate in the case where the decision boundary is smooth in some sense and is estimated using a series of successive quantile searches. Consider the boundary fragment class on $[0, 1]^d$ defined informally in [46] as the collection of sets in which the Bayes decision boundary is a Hölder smooth function of the first $d - 1$ coordinates. In $[0, 1]^2$, this implies that the boundary is crossed at most one time when traveling on a path along the second coordinate. The boundary can

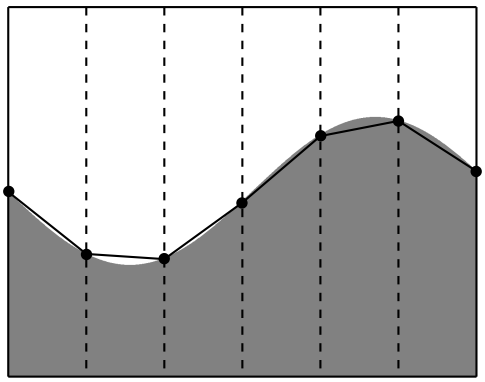

Figure 2.3: Example of set belonging to boundary fragment class and piecewise linear estimation of boundary.

be estimated by dividing the problem into strips along the first dimension, estimating the change point of each strip, and estimating the boundary as a piecewise linear function of the estimates, as shown in Fig. 2.3. For simplicity, we motivate the heuristics in this section by restricting f to the class of Lipschitz functions (a subset of Hölder smooth functions). Recall that a function $f : [0, 1]^d \to \mathbb{R}$ is said to be Lipschitz with constant $L \geq 0$ if for all $x_1 \neq x_2$

$$|f(x_1) - f(x_2)| \leq L \|x_1 - x_2\| .$$

Returning to Fig. 2.3, we see that a great deal of time would be wasted by returning to the origin after estimating the boundary at each strip. In this section, we leverage the assumed smoothness to intelligently initialize quantile search, resulting in significantly reduced sampling times, as shown in the simulations.

2.3.3.1 Initialization Using Previous Estimate

Assume we split the region of interest into K strips, each of which is a step function on the unit interval whose change point we wish to estimate. Let the true change point of the kth strip be θ^k and the estimate be $\hat{\theta}^k$. The smoothness assumption implies that θ^{k+1} is not "too far" from θ^k. For example, if f is Lipschitz with constant L and two successive strips are located at x_k and x_{k+1}, we know that $|\theta^k - \theta^{k+1}|/|x_k - x_{k+1}| \leq L$. For this reason, our first proposed improvement is to let the first sample location of the $k + 1$st strip X_0 be the previous estimate $\hat{\theta}^k$. Note that if we further assume a uniform prior on the subinterval allowed by the smoothness assumption, we are choosing our first sample as the minimum absolute error estimate, i.e., the median of the distribution. For later reference, we refer to this initialization as Improvement 1 (I-1). We show in Section 2.4 that this simple heuristic dramatically reduces the required sampling time of our algorithm.

2.3.3.2 Nonuniform Priors

Our second proposed algorithmic improvement involves assigning a nonuniform prior when beginning the search. Similar to the previous improvement, we utilize the function smoothness to assign lower starting probabilities to points unlikely to lie near the decision boundary. Letting $\hat{\theta}^k$ again be the boundary estimate at the kth strip, we assign a nonuniform prior whose mean is centered around $\hat{\theta}^k$. We propose the use of either a piecewise uniform or a Gaussian kernel function and refer to these as I-2.1 and I-2.2, respectively. Let the strip width $|x_k - x_{k+1}| = W$. We assign the prior probability for the $k+1$st strip to be either

$$\pi_0(x) = \begin{cases} c_1, & \left|x - \hat{\theta}^k\right| \leq LW \\ c_2, & \left|x - \hat{\theta}^k\right| > LW \end{cases} \qquad (\mathrm{I}-2.1),$$

where $c_1 > c_2$, or

$$\pi_0(x) = c_3 \exp\left(-\frac{(x-\hat{\theta}^k)^2}{2(LW)^2}\right) \qquad (\mathrm{I}-2.2),$$

where c_3 is a normalization constant so that the prior sums to 1. We discuss the choice of L and W in Section 2.4.

2.4 Simulations & Experiments

In this section, we show the efficacy of our algorithm through simulations. We first verify the theoretical guarantees provided in Section 2.3 and then compare the performance of PQS with the generalized version, which we refer to as TPQS. Next, we compare our method to proactive learning from [2]. We then show how a series of one-dimensional searches can be used to estimate the boundary of a two-dimensional hypoxic region in Lake Erie. We conclude with experimental results from Third Sister Lake in Ann Arbor, MI.

2.4.1 Verification of Algorithms

In this section, we verify through simulation the theoretical rate of convergence and distance traveled derived in Section 2.3.1. Further, we present simulated results for the PQS and TPQS algorithms and show that the desired tradeoff is achieved by both algorithms, with TPQS achieving better overall performance.

We first simulate the the DQS algorithm over a range of m from 2 to 20, where θ is swept over a 1000-point grid on the unit interval. The resulting average error after 20 samples is shown in the left-most plot of Fig. 2.4, while the average distance before convergence to an error of $\varepsilon = 1 \times 10^{-4}$

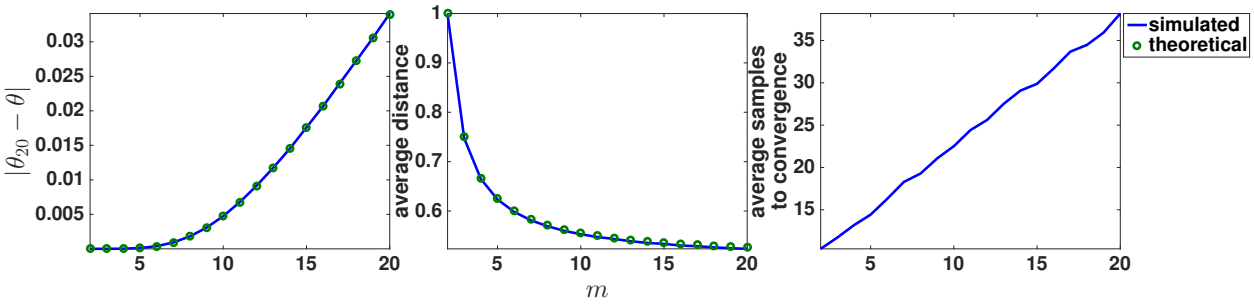

Figure 2.4: Simulated and theoretical values for DQS. Left-to-right: expected error after 20 samples, distance traveled before convergence to an estimation error less than 1×10^{-4}, simulated average samples required to converge to the same error.

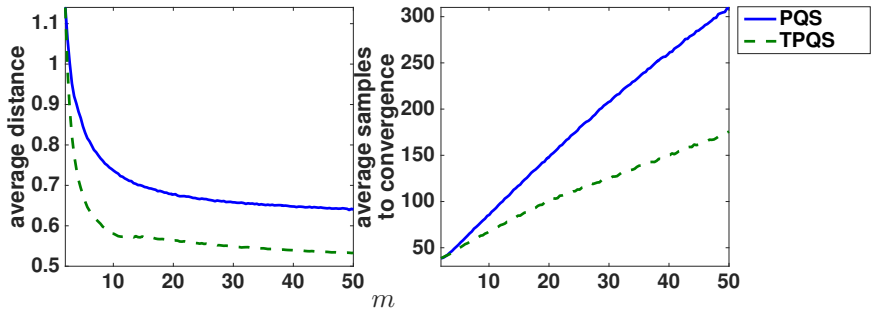

Figure 2.5: Average simulated values for PQS and TPQS. Left-to-right: distance traveled during estimation and number of samples required to converge.

is shown in the middle plot of the same figure. The figures show the theoretical values for expected error and distance match the simulated values. The right-most plot of Fig. 2.4 shows the number of samples required to converge to the same error. From the figures, our intuition is confirmed; the number of samples required is monotonically increasing in m, while the distance traveled is monotonically decreasing. This indicates that DQS achieves the desired tradeoff in the noise-free case.

Next, we simulate the PQS and TPQS algorithms with error probability $p = 0.1$ over a range of m from 2 to 50, where θ ranges over a 100-point grid on the unit interval with 100 random instances run for each value of θ. The left-hand plot of Fig. 2.5 shows the average number of samples required to converge to a mass of at least 0.9 at a single point, as described in Section 2.3.2.3. As in the deterministic case, the required number of samples increases monotonically with m. The right-hand plot of Fig. 2.5 shows the average distance traveled before converging to the same error value. Again, the distance decreases monotonically with m, indicating that the algorithm achieves the desired tradeoff in the noisy case. Further, we see that TPQS outperforms PQS both in terms of samples required and distance traveled. Because of this, we consider only TPQS in all remaining simulations.

Algorithm 2.3 Proactive Learning with Non-Uniform Costs [2] applied to one-dimensional threshold estimation

Input: search parameter $\lambda \in [0, 1]$, probability of error p
Initialize: $\pi_0(x) = 1$ for all $x \in [0, 1]$, $n \leftarrow 0$
while not converged **do**
  $X_{n+1} = \arg\max_x I(\theta; x, Y) - \lambda|X_n - x|$
  $Y_{n+1} \leftarrow f(X_{n+1}) \oplus U_{n+1}$, where $U_{n+1} \sim$ Ber(p)
  $\phi = \int_0^{X_{n+1}} \pi_n(x)dx$
  if $Y_{n+1} = 0$ **then**

$$\pi_{n+1}(x) = \begin{cases} \frac{(1-p)}{\phi * p}\pi_n(x) & x \leq X_{n+1} \\ \frac{p}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

  else

$$\pi_{n+1}(x) = \begin{cases} \frac{p}{\phi * p}\pi_n(x) & x \leq X_{n+1} \\ \frac{(1-p)}{\phi * p}\pi_n(x) & x > X_{n+1}. \end{cases}$$

  end if
  $n \leftarrow n + 1$
end while
estimate $\hat{\theta}_n$ such that $\int_0^{\hat{\theta}_n} \pi_n(x) = 1/2$

2.4.2 Application of Proactive Learning

The most competitive algorithm to quantile search is that of [2] applied to our problem. Of the scenarios explored in [2], the most relevant is Scenario 3, in which a non-uniform cost is charged for each label. The authors propose choosing each sample location to maximize the utility $U(X)$ at each round, where utility is defined as the difference between the value of the sample at X and the cost of taking that sample. The authors alternatively define utility as the ratio of value to cost, lending to a more natural interpretation that is similar to [63]. However, we found this version to result in poor performance, and hence we rely on the first approach. For comparison purposes, we maintain an estimate of the posterior distribution of θ as in quantile search. We define the value of a point X as the mutual information $I(\theta; X, Y)$ [38]. Note that in the noiseless case, Y is a deterministic function of θ, and hence mutual information is a misnomer. In this case, we still consider the reduction in entropy of θ given the measurement Y taken at point X. The relation of binary search to communicating a noisy sequence of bits over a binary symmetric channel has been well-studied [48]. In the noiseless case, we have

$$H(\theta) - H(\theta|X, Y) = H_b(X),$$

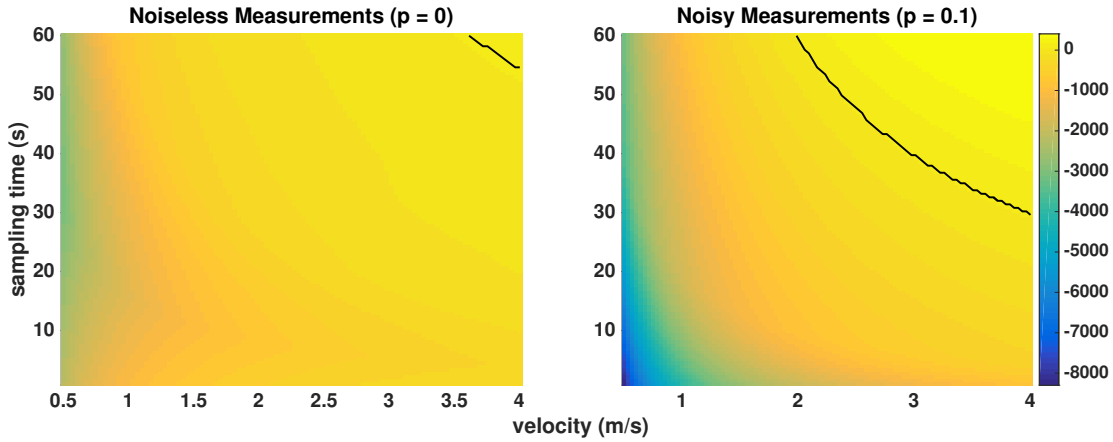

Figure 2.6: Difference in sampling time between quantile search and proactive learning under a variety of practical sampling regimes for both noiseless (left) and noisy (right) measurements with $p = 0.1$. Quantile search results in less required time for all points "southwest" of the black line.

where $H(\cdot)$ denotes the differential entropy and $H_b(\cdot)$ is the entropy of a Bernoulli random variable with corresponding probability X.

The noisy case of Section 2.3.2 corresponds to a binary symmetric channel with non-uniform priors [65]. In this case, we have

$$\begin{aligned} I(\theta; X, Y) &= H(\theta) - H(\theta|X, Y) \\ &= H_b(\phi * p) - H_b(p), \end{aligned}$$

where

$$\phi = \int_0^X \pi(x)dx.$$

Note that for $\phi = 1/2$ (i.e., PBS), the mutual information is $1 - H_b(p)$, which is the capacity of a noisy binary symmetric channel. We implement the proactive algorithm from [2], Eqn. (5) with two modifications in order to provide a fair comparison. First, the non-uniform cost in our case is the distance between the current location and the point under consideration, rather than the generic cost described in [2]. Second, we provide a tuning parameter that can be used similarly to m to balance between the number of samples and distance traveled during estimation. Pseudocode for this algorithm is given in Algorithm 2.3. In both the noiseless and noisy cases, we use the stopping criteria from DQS and PQS, respectively.

To obtain a profile of the performance of proactive learning, we simulate for both noiseless and noisy ($p = 0.1$) measurements, where we range λ over 100 points on the unit interval. We let θ range over a 100-point grid, and 200 random instances are run for each value of θ in the noisy case. To compare with DQS and TPQS, we simulate the average time required to perform sampling on the

unit interval under a variety of sampling times and travel times relevant to our problem of sampling in Lake Erie. We let the time per sample η in (2.5) range from 1-60 s per sample. For travel time, we consider a strip length of 40 km, about half the size of the central basin of Lake Erie, and let the velocity range from 0.5-4 m/s. Fig. 2.6 shows the difference in sampling time required by quantile search and proactive learning. The boundary of where quantile search outperforms proactive learning is shown in black, so that all points "up and to the right" of the boundary denote sampling regimes in which proactive learning requires less time than quantile search. The figure shows that in the majority of relevant cases, quantile search results in superior performance. However, in the case of large sampling time and high velocity, proactive learning generally performs better. Although not shown, we analyzed figures similar to Figs. 2.4 and 2.5 and saw that the number of samples required for proactive learning to converge reduces quickly with λ compared to quantile search, while the distance traveled reduces slowly. Thus, for scenarios in which sampling is significantly more costly than travel, proactive learning may be a more appropriate choice. This is likely due to the fact that proactive learning often takes comparatively large steps early in the measurement process, and investigating the properties of this algorithm is a topic for our future research.

2.4.3 Simulations on Lake Erie

In this section, we apply the quantile search and proactive learning algorithms to the problem of sampling hypoxic regions in Lake Erie. Fig. 2.7a shows the lake with an example hypoxic zone pictured in gray. In [46], the authors show that for the set of distributions such that the Bayes decision set is a boundary fragment, a variation on PBS can be used to estimate the boundary while achieving optimal rates up to a logarithmic factor. We now describe how the same approach can be used to estimate the hypoxic region in Lake Erie and demonstrate the benefits of our algorithm compared to PBS and proactive learning. The results in this section differ from our previous work [26] in that we consider a more realistic boundary derived from bathymetry data retrieved from [47]. To simulate the boundary of interest, we threshold the bathymetry data at a depth of 21 m and consider anything at a depth of greater than 21 m hypoxic. Although this may not be directly correlated with the hypoxic region, the resulting region is sufficiently irregular to test our algorithm and is visually similar to the regions found in [66]. Further, we previously considered only the time required to estimate the strips (described below) individually, whereas in this work we consider the entire sampling process.

Consider the instance of a hypoxic region shown in Fig. 2.7a. Using models and measurements from previous years (e.g., historical data from [66]), it is reasonable to assume we can split the lake into intervals so that the boundary does not significantly violate the boundary fragment assumption. Splitting the lake along the line $y = b$ yields the two sets above and below the dashed line in

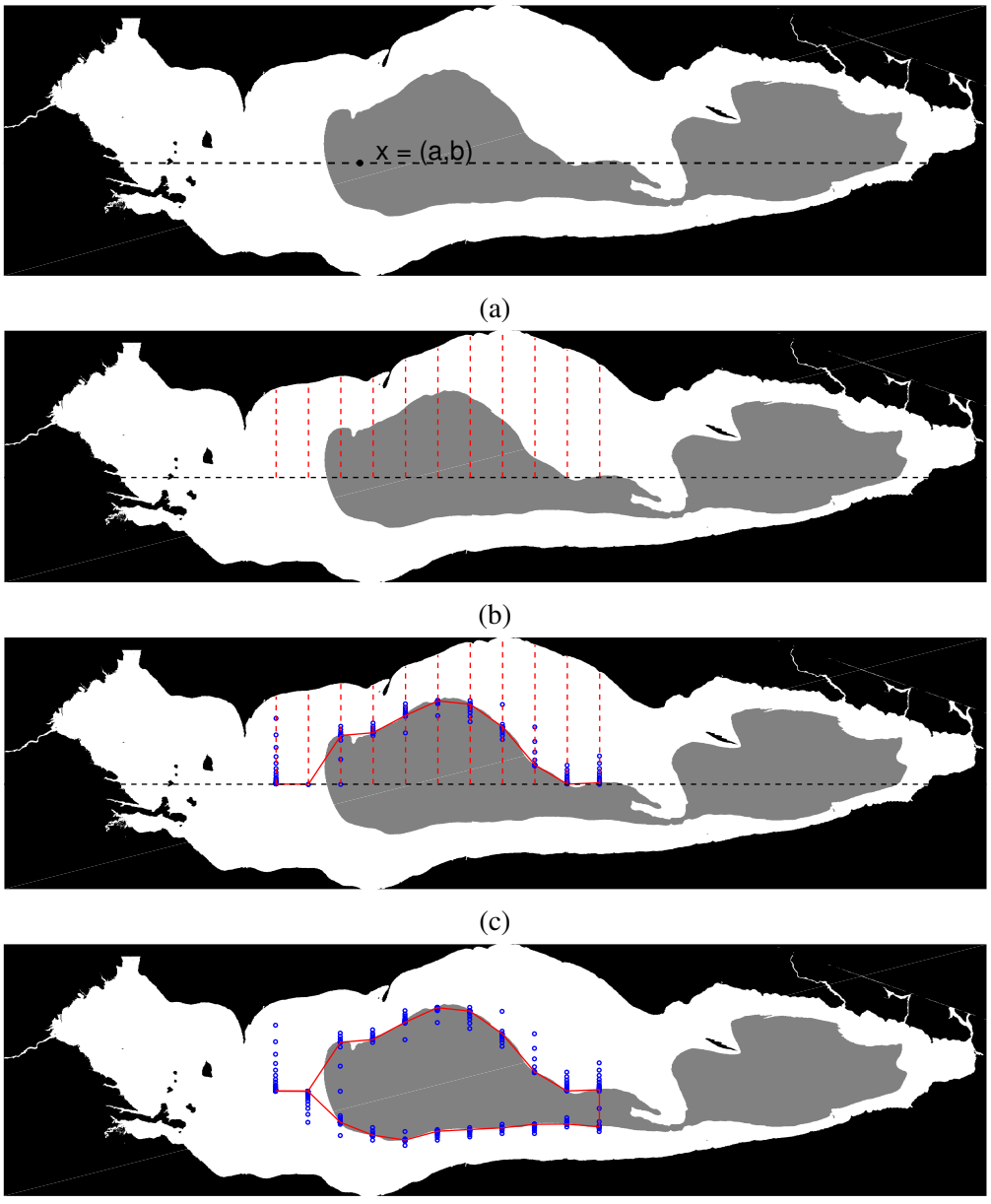

Figure 2.7: Proposed sampling procedure for detection of hypoxic region in Lake Erie. (a) Lake Erie with hypoxic region illustrated in gray and split along $x = (a, b)$. (b) Division of top portion into strips. (c) Estimation procedure for top of lake with sample locations shown in blue and estimated boundary in solid red. (d) Final sample locations and estimation of entire boundary.

Fig. 2.7a. Now we can further divide the problem into strips along the first dimension, as shown by the solid red line in Fig. 2.7b. Along each of these strips, the problem reduces to change point estimation of a one-dimensional threshold classifier as we have studied thus far. After estimating the change point at each strip, the boundary is estimated as a piecewise linear function of the estimates, as shown in Fig. 2.7c. The same procedure is used for the bottom portion of the lake, with the final

Sampling Scenarios	Sampling Time (s) Velocity (m/s)	60 4	60 0.5	10 4	10 0.5
Sampling Parameters	m λ	9.48 0.17	43.00 0.13	43.00 0.13	43.00 0.10
Base Algorithm without Improvements	Bisection DQS Proactive Learning	2.14 2.62 2.84	15.96 19.97 21.20	2.00 2.52 2.67	15.82 19.44 20.45
I-1	Bisection DQS Proactive Learning	1.99 **1.44** 1.47	14.86 **9.41** 9.99	1.86 **1.19** 1.26	14.73 **9.09** 9.62

Table 2.1: Total sampling time (in days) for various search methods under noiseless measurements and a variety of sampling times and velocities. Fastest time for each scenario shown in bold.

estimation shown in Fig. 2.7d. In all cases, we choose the optimal m by estimating the average number of samples and distance traveled via simulations and note the chosen value in the tables.

We apply this procedure to the hypoxic region shown in Fig. 2.7a using 11 strips for a variety of values for time per sample and speed of watercraft. To simulate an actual sampling pattern, we proceed counterclockwise through the strips, beginning from the top left, and record the total distance traveled and number of samples taken. We consider several sampling strategies. As a baseline, we use binary bisection with no algorithmic improvements, i.e., quantile search with fixed $m = 2$. We also show DQS with a fixed m chosen to optimize the total sampling time using the average scale factor for the entire lake. Next, we show the sampling time for proactive learning with λ chosen similarly. Finally, we consider these scenarios while employing Improvement 1, where we initialize our search algorithm using the previous boundary estimate. We forego the application of I-2.1 and I-2.2, as they will have minimal impact in the noiseless case. Table 2.1 shows the resulting sampling time (in days) required to estimate the boundary of the hypoxic region. When I-1 is not in use, binary search outperforms our algorithm. This is due to the fact that the craft must travel back to the position $1/m$ at each strip, a significant distance when m is small and the boundary estimate is not near the origin. However, this problem is overcome by employing I-1, in which case quantile search requires roughly half the sampling time required by binary search. Further, DQS outperforms proactive learning, even in the scenarios where DQS requires more time on a single strip. In the case of low sampling time and low velocity, we see that DQS significantly outperforms proactive learning, which matches our expectations based on Fig. 2.6.

Next, we apply the same procedure to the noisy case with a probability of measurement error $p = 0.1$ averaged over 100 random instances. Due to the performance benefits shown in DQS, we employ I-1 in all sampling scenarios. For both I-2.1 and I-2.2, we choose the strip width W based on the number of strips, which is a function of the desired estimation error. Choosing W small will result in more accurate estimation but require more sampling time. In practice one would estimate

Sampling Scenarios	Sampling Time (s) Velocity (m/s)	60 4	60 0.5	10 4	10 0.5
Sampling Parameters	m λ	6.40 0.30	11.17 0.29	10.80 0.29	62.16 0.29
I-1	PBS	2.64	19.00	2.38	18.61
	TPQS	1.83	10.84	1.38	9.57
	Proactive Learning	**1.72**	11.67	1.48	11.47
I-1, I-2.1	PBS	2.63	18.66	2.37	18.66
	TPQS	1.82	10.85	1.38	9.58
	Proactive Learning	1.73	11.69	1.47	11.44
I-1, I-2.2	PBS	2.58	18.30	2.33	18.11
	TPQS	1.83	**10.75**	**1.37**	**9.56**
	Proactive Learning	1.73	11.77	1.49	11.52

Table 2.2: Total sampling time (in days) for various search methods under noisy measurements with $p = 0.1$ and a variety of sampling times and velocities. Fastest time for each scenario shown in bold.

L using historical data. We estimate L numerically as

$$\hat{L} = \arg\max_i \frac{|f(x_i) - f(x_i + \delta)|}{\delta},$$

where we choose δ to be $0.1W$ to prevent the value of L from being inflated by a single point in f with high derivative. Note that since we are only using L to generate priors for our search function, even an aggressive choice will not prevent our algorithm from finding the true boundary. In some cases, where the function is very smooth in many places and has high derivative in a few places, a user may wish to choose L smaller than the estimated value to reduce sampling time. In I-2.1, we choose c_1 and c_2 such that the probability within LW of $\hat{\theta}^k$ is 100 times the probability outside this region, i.e., $c_1 = 100c_2$.

Table 2.2 shows the resulting sampling times under the various sampling scenarios. The results of the table across all sampling parameters indicate that TPQS with a Gaussian prior is the best sampling strategy in most cases. In the case of a 60 sec sampling time and 4 m/s velocity, proactive learning outperforms our algorithm, which is consistent with the results of Fig. 2.6. Interestingly, the use of nonuniform priors results in a small benefit in most cases. This is likely due to the fact that the bottom half of the hypoxic region is extremely smooth, and hence our value of L is not aggressive enough for these strips. A better choice may be to choose L separately for the strips on top and bottom of the lake.

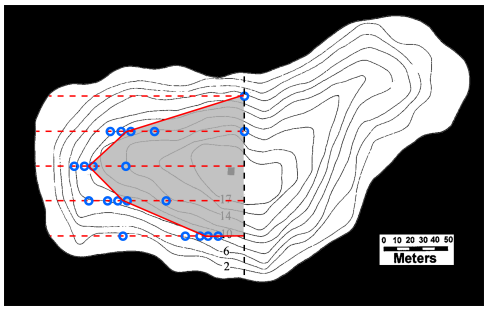

Figure 2.8: Delineated hypoxic region on the western half of Third Sister Lake.

2.4.4 Experiments on Third Sister Lake

In this section, we present the implementation and performance of the DQS algorithm in the field. The algorithm was tested on a robotic boat that was deployed at Third Sister Lake in Ann Arbor, Michigan. Third Sister Lake is a spring-fed kettle lake with an area of 9.4 acres and a maximum depth of 17 meters that notably exhibits hypoxic conditions on an annual basis [67]. The smaller size and calmer waters of Third Sister Lake posed an ideal test bed for evaluating the algorithm. Because of the high fidelity of the oxygen sensor used, only the DQS algorithm was tested in the field. Further, these experiments are intended as a pilot study to motivate the use of our algorithm in larger bodies of water, such as Lake Erie.

The robotic boat platform [68] features an Android cellular phone for GPS navigation and 3G cellular communications. The prevalent cell coverage at Third Sister Lake enables bi-directional communication with the boat for remotely tracking and delineating the evolution of the hypoxic region in real-time. For a given GPS coordinate, the boat autonomously navigates to the destination to collect a sample. The platform was outfitted with a motorized winch to raise and lower a suite of water quality sensors to measure dissolved oxygen throughout the water column at each sampling location. Due to the low noise level of these sensors, we employed the noiseless version of the algorithm with I-1.

Leveraging the persistent Internet connectivity of the robotic boat, the platform was paired with a web-service-based cyberinfrastructure [69]. This enabled the same script used to develop the algorithm to be tested in the field by modifying the script to open a web connection and directly control the boat. Time-stamped location and measurement data were immediately accessible to the algorithm to direct where the boat should sample next. Taking a web-based approach provides the flexibility more readily interface with any web-enabled robotic boat that may be more suitable for increased winds and waves of more challenging sites.

We present the results from a sampling campaign on November 17, 2015 in Fig. 2.8. Third Sister Lake was divided into five horizontal strips, along which an average of five samples were taken until GPS precision could no longer distinguish between two locations. The estimated velocity

of the robotic boat was 0.1 m/s. Due to the need to lower and raise the winch for each sample location, the average time to collect a sample was 300 s, resulting in an optimal sampling parameter of $m = 2$. We observed that over the course of five hours, the platform successfully identified and delineated the hypoxic zone as directed by the algorithm. In comparison, a uniform sampling at the same resolution would take an estimated 27 hours. The successful results from the experiments on Third Sister Lake demonstrate the potential to extend this algorithm to other lake systems including Lake Erie.

2.5 Conclusion

We have presented an active learning algorithm for spatial sampling capable of balancing the number of samples and distance traveled in order to minimize the overall sampling time. To the best of our knowledge, this is the only nonuniformly penalized active learning algorithm accompanied by theoretical guarantees. We have shown how our algorithm can be used to estimate a two-dimensional region of hypoxia under certain smoothness assumptions on the boundary, and empirical results indicate the benefits of quantile search over traditional binary search as well as other active learning methods in the literature.

Several open questions remain. Deriving or bounding the expected distance for the TPQS algorithm is an important next step. The boundary fragment class mentioned here is restrictive [6], and the extension to more general cases would be of interest. As a first step toward this analysis, it would be interesting to study how the algorithm behaves under slight deviations from the boundary fragment class assumptions. The recent work of [70] describes a graph-based algorithm that employs PBS in a method for higher-dimensional nonparametric estimation. Extending this idea to penalize distance traveled is a promising avenue for practical applications of quantile search. The PQS algorithm requires knowledge of the noise parameter p in order to update the posterior. The algorithms presented in [33, 71] enjoy the property that they are adaptive to unknown noise levels. The development of a noise-adaptive probabilistic search would certainly be of great interest, with potential applications in areas such as stochastic optimization [71] beyond direct applicability to this problem. Finally, while we have exactly characterized both the distance traveled and samples required for DQS, it is not clear whether our algorithm is optimal in any sense. The optimality analysis is difficult, as the worst-case distance traveled is always one, and hence standard approaches to minimax bounds are not applicable.

CHAPTER 3

Active Learning for Subspace Clustering

3.1 Introduction

The union of subspaces (UoS) model, in which data vectors lie near one of several subspaces, has been used actively in the computer vision community on datasets ranging from images of objects under various lighting conditions [7] to visual surveillance tasks [72]. The recent textbook [73] includes a number of useful applications for this model, including lossy image compression, clustering of face images under different lighting conditions, and video segmentation. Subspace clustering algorithms utilize the UoS model to cluster data vectors and estimate the underlying subspaces, achieving excellent performance on a variety of real datasets. However, as we will show in Section 3.4, even oracle UoS classifiers do not achieve perfect clustering on these datasets. While current algorithms for subspace clustering are unsupervised, in many cases a human could provide relevant information in the form of pairwise constraints between points, e.g., answering whether two images are of the same person or whether two objects are the same.

The incorporation of pairwise constraints into clustering algorithms is known as pairwise-constrained clustering (PCC). PCC algorithms use supervision in the form of *must-link* and *cannot-link* constraints by ensuring that points with must-link constraints are clustered together and points with cannot-link constraints are clustered apart. In [12], the authors investigate the phenomenon that incorporating poorly-chosen constraints can lead to an increase in clustering error, rather than a decrease as one would expect from additional label information. This is because points constrained to be in the same cluster that are otherwise dissimilar can confound the constrained clustering algorithm. For this reason, researchers have turned to *active* query selection methods, in which constraints are intelligently selected based on a number of heuristics. Active methods such as [1] have been shown to significantly reduce clustering error with a modest number of pairwise constraints, and in [74] the authors receive constraints from people with no special training via Amazon Mechanical Turk. These algorithms perform well across a number of datasets but do not take advantage of any known structure in the data. In the case where data lie on a union of

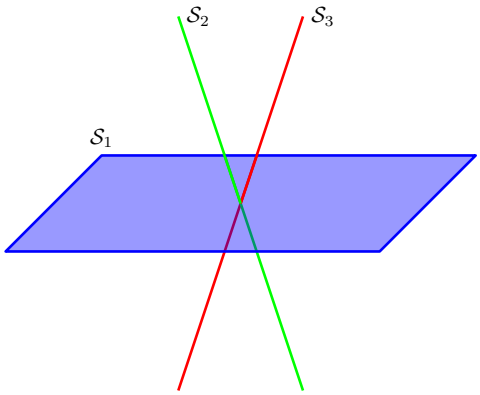

Figure 3.1: Example union of $K = 3$ subspaces of dimensions $d_1 = 2$, $d_2 = 1$, and $d_3 = 1$.

subspaces, one would hope that knowledge of the underlying geometry could give hints as to which points are likely to be clustered incorrectly.

Let $\mathcal{X} = \left\{x_i \in \mathbb{R}^D\right\}_{i=1}^N$ be a set of data points lying near a union of K linear subspaces of the ambient space. We denote the subspaces by $\{\mathcal{S}_k\}_{k=1}^K$, each having dimension d_k. An example union of subspaces is shown in Fig. 3.1, where $d_1 = 2$, $d_2 = d_3 = 1$. The goal of subspace clustering algorithms has traditionally been to cluster the points in $\mathcal{X}$ according to their nearest subspace without any supervised input. We turn this around and ask whether this model is useful for active clustering, where we request a very small number of intelligently selected labels. A key observation when considering data well-modeled by a union of subspaces is that uncertain points will be ones lying equally distant to multiple subspaces. Using a novel definition of margin tailored for the union of subspaces model, we incorporate this observation into an active subspace clustering algorithm.

Our contributions are as follows. We introduce a novel algorithm for pairwise constrained clustering that leverages UoS structure in the data. A key step in our algorithm is choosing points of *minimum margin*, i.e., those lying near a decision boundary between subspaces. We define a notion of margin for the UoS model and provide theoretical insight as to why points of minimum margin are likely to be misclustered by unsupervised algorithms. We show through extensive experimental results that when the data lie near a union of subspaces, our method drastically outperforms existing PCC algorithms, requiring far fewer queries to achieve perfect clustering. Our datasets range in dimension from 256-2016, number of data points from 320-9298, and number of subspaces from 5-100. On ten MNIST digits with a modest number of queries, we get 5% classification error with only 500 pairwise queries compared to about 20% error for current state-of-the-art PCC algorithms and 35% for unsupervised algorithms. We also achieve 0% classification error on the full Yale, COIL, and USPS datasets with a small fraction of the number of queries needed by competing algorithms. In datasets where we do not expect subspace structure, our algorithm still achieves competitive performance. Further, our algorithm is agnostic to the input subspace clustering algorithm and can therefore take advantage of any future algorithmic advances for subspace clustering.

3.2 Related Work

A survey of recently developed subspace clustering algorithms can be found in [75] and the textbook [73]. In these and more recent work, clustering algorithms that employ spectral methods achieve the best performance on most datasets. Notable examples of such algorithms include Sparse Subspace Clustering (SSC) [23] and its extensions [76, 77], Low-Rank Representation (LRR) [78], Thresholded Subspace Clustering (TSC) [41], and Greedy Subspace Clustering (GSC) [40]. Many recent algorithms exist with both strong theoretical guarantees and empirical performance, and a full review of all approaches is beyond the scope of this work. However, the core element of all recent algorithms lies in the formation of the affinity matrix, after which spectral clustering is performed to obtain label estimates. In SSC, the affinity matrix is formed via a series of ℓ_1-penalized regressions. LRR uses a similar cost function but penalizes the nuclear norm instead of the ℓ_1. TSC thresholds the spherical distance between points, and GSC works by successively (greedily) building subspaces from points likely to lie in the same subspace. Of these methods, variants of SSC achieve the best overall performance on benchmark datasets and has the strongest theoretical guarantees, which were introduced in [23] and strengthened in numerous recent works [43, 79, 80, 81]. TSC and GSC also provide theoretical guarantees and are significantly less computationally-demanding than SSC, though this challenge is overcome by SSC variants relying on orthogonal matching pursuit [76] and the elastic net framework [77]. While the development of efficient algorithms with stronger guarantees has received a great deal of attention, very little attention has been paid to the question of what to do about data that cannot be correctly clustered. As we illustrate in Section 3.4, even oracle PCA classifiers result in some clustering error for common datasets. Thus, when reducing clustering error to zero (or near zero) is a priority, users must look beyond unsupervised subspace clustering algorithms to alternative methods. One such method is to request some supervised input in the form of pairwise constraints, leading to the study of pairwise-constrained clustering (PCC).

PCC algorithms work by incorporating *must-link* and *cannot-link* constraints between points, where points with must-link constraints are forced (or encouraged in the case of spectral clustering) to be clustered together, and points with cannot-link constraints are forced to be in separate clusters. In many cases, these constraints can be provided by a human labeler. For example, in [74], the authors perform experiments where comparisons between human faces are provided by users of Amazon Mechanical Turk with an error rate of 1.2%. Similarly, for subspace clustering datasets such as Yale B and MNIST, a human could easily answer questions such as, "Are these two faces the same person?" and "Are these two images the same number?" An early example of PCC is found in [82], where the authors modify the K-means cost function to incorporate such constraints. In [83], the authors utilize active methods to initialize K-means in an intelligent "EXPLORE" phase, during which neighborhoods of must-linked points are built up. After this phase, new points are queried

against representatives from each neighborhood until a must-link is obtained. A similar explore phase is used in [84], after which a min-max approach is used to select the most uncertain sample. Early work on constrained spectral clustering appears in [85, 86], in which spectral clustering is improved by examining the eigenvectors of the affinity matrix in order to determine the most informative points. However, these methods are limited to the case of two clusters and therefore impractical in many cases.

More recently, the authors in [1, 74] improve constrained clustering by modeling which points will be most informative given the current clustering, with state-of-the-art results achieved on numerous datasets by the algorithm in [1], referred to as Uncertainty Reducing Active Spectral Clustering (URASC). URASC works by maintaining a set of *certain sets*, whereby points in the same certain set are must-linked and points in different certain sets are cannot-linked. A test point x_T is selected via an uncertainty-reduction model motivated by matrix perturbation theory, after which queries are presented in an intelligent manner until x_T is either matched with an existing certain set or placed in its own new certain set. In practice [87], the certain sets are initialized using the EXPLORE algorithm of [83].

While we are certainly not the first to consider actively selecting labels to improve clustering performance, to the best of our knowledge we are the first to do so with structured clusters. Structure within and between data clusters is often leveraged for unsupervised clustering [88], and that structure is also leveraged for adaptive sampling of the structured signals themselves (e.g., see previous work on sparse [89, 90], structured sparse [91], and low rank signals [92]). This paper emphasizes the power of that structure for reducing the number of required labels in an active learning algorithm as opposed to reducing the number of samples of the signal itself, and points to exciting open questions regarding the tradeoff between signal measurements and query requirements in semi-supervised clustering.

3.3 UoS-Based Pairwise-Constrained Clustering

Recall that $\mathcal{X} = \left\{x_i \in \mathbb{R}^D\right\}_{i=1}^N$ is a set of data points lying on a union of K subspaces $\{\mathcal{S}_k\}_{k=1}^K$, each having dimension d. In this work, we assume all subspaces have the same dimension, but it is possible to extend our algorithm to deal with non-uniform dimensions. The goal is to cluster the data points according to this generative model, i.e., assigning each data point to its (unknown) subspace. In this section we describe our algorithm, which actively selects pairwise constraints in order to improve clustering accuracy. The key step is choosing an informative query test point, which we do using a novel notion of *minimum subspace margin*.

Denote the true clustering of a point $x \in \mathcal{X}$ by $C(x)$. Let the output of a clustering algorithm

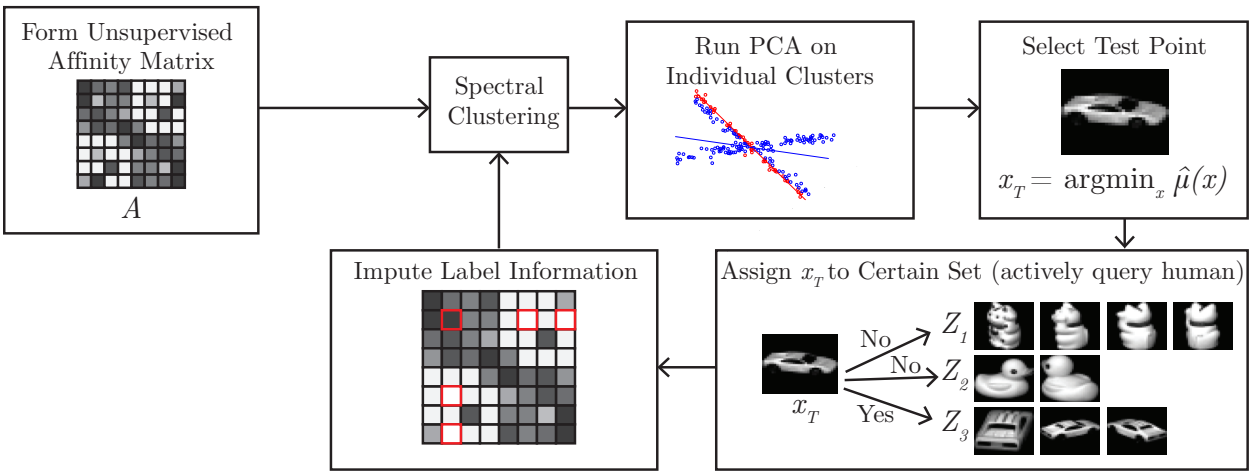

Figure 3.2: Diagram of SUPERPAC algorithm for pairwise constrained clustering.

(such as SSC) be an affinity/similarity matrix A and a set of label estimates $\left\{\hat{C}(x_i)\right\}_{i=1}^{N}$. These are the inputs to our algorithm. The high-level operation of our algorithm is as follows. To initialize, we build a set of certain sets $\mathcal{Z}$ using an EXPLORE-like algorithm similar to that of [83]. Certain sets are in some sense equivalent to labels in that points within a certain set belong to the same cluster and points across certain sets belong to different clusters. Following this, the following steps are repeated until a maximum number of queries has been made:

1. **Spectral Clustering:** Obtain label estimates via spectral clustering.

2. **PCA on each cluster:** Obtain a low-dimensional subspace estimate from points currently sharing the same estimated cluster label.

3. **Select Test Point:** Obtain a test point x_T using subspace margin with respect to the just estimated subspaces.

4. **Assign x_T to Certain Set:** Query the human to compare the test point with representatives from certain sets until a must-link is found or all certain sets have been queried, in which case the test point becomes its own certain set.

5. **Impute Label Information:** Certain sets are used to impute must-link and cannot-link values in the affinity matrix.

We refer to our algorithm as SUPERPAC (SUbsPace clustERing with Pairwise Active Constraints). A diagram of the algorithm is given in Fig. 3.2, and we outline each of these steps below and provide pseudocode in Algorithm 3.1.

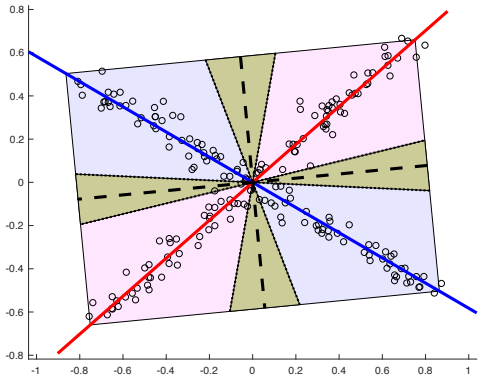

Figure 3.3: Illustration of subspace margin. The blue and red lines are the generative subspaces, with corresponding disjoint decision regions. The yellow-green color shows the region within some margin of the decision boundary, given by the dotted lines.

3.3.1 Sample Selection via Margin

Min-margin points have been studied extensively in active learning; intuitively, these are points that lie near the decision boundary of the current classifier. In [37], the author notes that actively querying points of minimum margin (as opposed to maximum entropy or minimum confidence) is an appropriate choice for reducing classification error. In [33], the authors present a margin-based binary classification algorithm that achieves an optimal rate of convergence (within a logarithmic factor).

In this section, we define a novel notion of margin for the UoS model and provide theoretical insight as to why points of minimum margin are likely to be misclustered. For a subspace $\mathcal{S}_k$ with orthonormal basis U_k, let the distance of a point to that subspace be $\text{dist}(x, \mathcal{S}_k) = \min_{y \in \mathcal{S}_k} \|x - y\|_2 = \left\|x - U_k U_k^T x\right\|_2$. Let $k^* = \arg\min_{k \in [K]} \text{dist}(x, \mathcal{S}_k)$ be the index of the closest subspace, where $[K] = \{1, 2, \cdots, K\}$. Then the subspace margin of a point $x \in \mathcal{X}$ is the ratio of closest and second closest subspaces, defined as

$$\hat{\mu}(x) = 1 - \max_{j \neq k^*, j \in [K]} \frac{\text{dist}(x, S_{k^*})}{\text{dist}(x, S_j)}. \tag{3.1}$$

The point of minimum margin is then defined as $\arg\min_{x \in \mathcal{X}} \hat{\mu}(x)$. Note that the fraction is a value in $[0, 1]$, where the a value of 0 implies that the point x is equidistant to its two closest subspaces. This notion is illustrated in Figure 3.3, where the yellow-green color shows the region within some margin of the decision boundary.

In the following theorem, we show that points lying near the intersection of subspaces are included among those of minimum margin with high probability. This method of point selection is then motivated by the fact that the difficult points to cluster are those lying near the intersection of subspaces [12]. Further, theory for SSC ([11],[15]) shows that problematic points are those having

large inner product with some or all directions in other subspaces. Subspace margin captures exactly this phenomenon.

Theorem 3.1. *Consider two d-dimensional subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. Let $y = x + n$, where $x \in \mathcal{S}_1$ and $n \sim \mathcal{N}(0, \sigma^2 I_D)$. Define*

$$\mu(y) = 1 - \frac{\operatorname{dist}(y, \mathcal{S}_1)}{\operatorname{dist}(y, \mathcal{S}_2)} \, .$$

Then

$$1 - \frac{(1+\varepsilon)\sqrt{\sigma^2(D-d)}}{(1-\varepsilon)\sqrt{\sigma^2(D-d) + \operatorname{dist}(x, \mathcal{S}_2)^2}} \leq \mu(y) \leq 1 - \frac{(1-\varepsilon)\sqrt{\sigma^2(D-d)}}{(1+\varepsilon)\sqrt{\sigma^2(D-d) + \operatorname{dist}(x, \mathcal{S}_2)^2}},$$

The proof is given in Appendix B. Note that if $\operatorname{dist}(y, \mathcal{S}_1) \leq \operatorname{dist}(y, \mathcal{S}_2)$, then $\mu(y) = \hat{\mu}(y)$. In this case, Thm. 3.1 states that under the given noise model, points with small residual to the incorrect subspace (i.e., points near the intersection of subspaces) will have small margin. These are exactly the points for which supervised label information will be most beneficial.

The statement of Thm. 3.1 allows us to quantify exactly how near a point must be to the intersection of two subspaces to be considered a point of minimum margin. Let $\phi_1 \leq \phi_2 \leq \cdots \leq \phi_d$ be the d principal angles[1] between $\mathcal{S}_1$ and $\mathcal{S}_2$. If the subspaces are very far apart, $\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$ is near 1, and if they are very close $\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$ is near zero. Note that, for any $x \in \mathcal{S}_1$,

$$\sin^2(\phi_1) \leq \operatorname{dist}(x, \mathcal{S}_2)^2 \leq \sin^2(\phi_d) \, ;$$

that is, there are bounds on $\operatorname{dist}(x, \mathcal{S}_2)$ depending on the relationship of the two subspaces. We also know that if x is drawn using isotropic Gaussian weights from a basis for $\mathcal{S}_1$, then

$$\mathbb{E}\left[\operatorname{dist}(x, \mathcal{S}_2)^2\right] = \frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i) \, .$$

Given this, we might imagine that margin of the noisy points is a useful indicator of points near the intersection in a scenario where $\sin^2(\phi_1)$ is small but $\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$ is not, e.g., when the subspaces have an intersection but are distant in other directions. With this in mind we state the following corollary, whose proof can be found in Appendix B.

Corollary 3.1. *Suppose $x_1 \in \mathcal{S}_1$ is such that*

$$\operatorname{dist}(x_1, \mathcal{S}_2)^2 = \sin^2(\phi_1) + \delta\left(\frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)\right) \tag{3.2}$$

[1] See [42] for a definition of principal angles.

for some small $\delta \geq 0$; that is, x_1 is close to the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. Let x_2 be a random point in $\mathcal{S}_1$ generated as $x_2 = U_1 w$ where U_1 is a basis for $\mathcal{S}_1$ and $w \sim \mathcal{N}(0, \frac{1}{d} I_d)$. We observe $y_i = x_i + n_i$, where $n_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, 2$. If there exists $\tau > 1$ such that

$$\delta < \frac{5}{7} - \frac{1}{\tau}$$

and

$$\tau \left(\sin^2(\phi_1) + \frac{1}{6}\sigma^2 (D - d) \right) < \frac{1}{d} \sum_{i=1}^{d} \sin^2(\phi_i) \ , \tag{3.3}$$

that is, the average angle is sufficiently larger than the smallest angle, then

$$\mathbb{P}\left\{ \mu(y_1) < \mu(y_2) \right\} \geq 1 - e^{-c\left(\frac{7}{100}\right)^2 ds} - 4e^{-c\left(\frac{1}{50}\right)^2 (D-d)}$$

where $\mu(y)$ is defined as in Thm. 3.1, c is an absolute constant, and $s = \frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i)$.

We make some remarks first to connect our results to other subspace distances that are often used. Perhaps the most intuitive form of subspace distance between that spanned by U_1 and U_2 is $\frac{1}{d}\|(I - U_1 U_1)^T U_2\|_F^2$; if the two subspaces are the same, the projection onto the orthogonal complement is zero; if they are orthogonal, we get the norm of U_2 alone, giving a distance of 1. This is equal to the more visually symmetric $1 - \frac{1}{d}\|U_1^T U_2\|_F^2$, another common distance. Further we note that, by the definition of principal angles [42],

$$1 - \frac{1}{d}\|U_1^T U_2\|_F^2 = 1 - \frac{1}{d}\sum_{i=1}^{d} \cos^2(\phi_i) = \frac{1}{d}\sum_{i=1}^{d} \sin^2(\phi_i) \ .$$

From Equation (3.2), we see that the size of δ determines how close $x_1 \in \mathcal{S}_1$ is to $\mathcal{S}_2$; if $\delta = 0$, x_1 is as close to $\mathcal{S}_2$ as possible. For example, if $\phi_1 = 0$, the two subspaces intersect, and $\delta = 0$ implies that $x_1 \in \mathcal{S}_1 \cap \mathcal{S}_2$. Equation (3.3) captures the gap between average principal angle and the smallest principal angle. We conclude that if this gap is large enough and δ is small enough so that x_1 is close to $\mathcal{S}_2$, then the observed y_1 will have smaller margin than the average point in $\mathcal{S}_1$, even when observed with noise.

For another perspective, consider that in the noiseless case, for $x_1, x_2 \in \mathcal{S}_1$, the condition $\mathrm{dist}(x_1, \mathcal{S}_2) < \mathrm{dist}(x_2, \mathcal{S}_2)$ is enough to guarantee that x_1 lies nearer to $\mathcal{S}_2$. Under the given additive noise model ($y_i = x_i + n_i$ for $i = 1, 2$) the gap between $\mathrm{dist}(x_1, \mathcal{S}_2)$ and $\mathrm{dist}(x_2, \mathcal{S}_2)$ must be larger by some factor depending on the noise level. After two applications of Thm. 3.1 and rearranging terms, we have that $\mu(y_1) < \mu(y_2)$ with high probability if

$$\beta \mathrm{dist}(x_2, \mathcal{S}_2)^2 - \mathrm{dist}(x_1, \mathcal{S}_2)^2 > (1 - \beta)\sigma^2 (D - d). \tag{3.4}$$

Algorithm 3.1 SUPERPAC

Input: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$: data, K: number of clusters, d: subspace dimension, A: affinity matrix, maxQueries: maximum number of pairwise comparisons
Estimate Labels: $\hat{C} \leftarrow$ SPECTRALCLUSTERING(A,K)
Initialize Certain Sets: Initialize $\mathcal{Z} = \{Z_1, \cdots, Z_{n_c}\}$ and numQueries via UOS-EXPLORE Algorithm 3.2
while numQueries $<$ maxQueries **do**
PCA on Each Cluster: Solve

$$\mathcal{S}_k = \min_{U \in \mathbb{R}^{D \times d}} \sum_{i:\hat{C}(x_i)=k} \|x_i - UU'x_i\|^2 .$$

Obtain Test Point: select $x_T \leftarrow \arg\min_{x \in \mathcal{X}} \hat{\mu}(x)$
Assign x_T to Certain Set:

Sort $\{Z_1, \cdots, Z_{n_c}\}$ in order of most likely must-link (via subspace residual for x_T), query x_T against representatives from Z_k until must-link constraint is found or $k = n_c$. If no must-link constraint is found, set $\mathcal{Z} \leftarrow \{Z_1, \cdots, Z_{n_c}, \{x_T\}\}$ and increment n_c.

Impute Constraints: Set $A_{ij} = A_{ji} = 1$ for (x_i, x_j) in the same certain set and $A_{ij} = A_{ji} = 0$ for (x_i, x_j) in different certain sets (do not impute for points absent from certain sets).
Estimate Labels: $\hat{C} \leftarrow$ SPECTRALCLUSTERING(A,K)
end while

where $\beta = ((1-\varepsilon)/(1+\varepsilon))^4$, a value near 1 for small ε. Equation (3.4) shows that the gap $\text{dist}(x_2, \mathcal{S}_2)^2 - \text{dist}(x_1, \mathcal{S}_2)^2$ must grow (approximately linearly) with the noise level σ^2. The relationship of this gap to the subspace distances is quantified by Corollary 3.1; plugging $\sin^2(\phi_1)$ from Equation (3.2) into Equation (3.3) and rearranging yields a statement of the form in Equation (3.4).

3.3.2 Pairwise Constrained Clustering with SUPERPAC

We now describe SUPERPAC in more detail, our algorithm for PCC when data lie near a union of subspaces, given in Algorithm 3.1. The algorithm begins by initializing a set of disjoint certain sets, an optional process described in the following section. Next our algorithm assigns the points most likely to be misclassified to certain sets by presenting a series of pairwise comparisons. Finally, we impute values onto the affinity matrix for all points in the certain sets and perform spectral clustering. The process is then repeated until the maximum number of pairwise comparisons has been reached.

Let x_T be the test point chosen as the min-margin point. Our goal is to assign x_T to a certain set using as the fewest number of queries possible. For each certain set Z_k, the representative x_k is chosen as the maximum-margin point within the set. Next, for each k, we let U_k be the d-

dimensional PCA estimate of the matrix whose columns are the points $\left\{x \in \mathcal{X} : \hat{C}(x) = \hat{C}(x_k)\right\}$. We then query our test point x_T against the representatives x_k in order of residual $\left\|x_T - U_k U_k^T x_T\right\|_2$ (smallest first). If a must-link constraint is found, we place x_T in the corresponding certain set. Otherwise, we place x_T in its own certain set and update the number of certain sets. Pseudocode for the complete algorithm is given in Algorithm 3.1. As a technical note, we first normalize the input affinity matrix A so that the maximum value is 2. For must-link constraints, we impute a value of 1 in the affinity matrix, while for cannot-link constraints we impute a 0. The approach of imputing values in the affinity matrix is common in the literature but does not strictly enforce the constraints. Further, we found in our experiments that imputing the maximum value in the affinity matrix resulted in unstable results. Thus, users must be careful to not only choose the correct constraints as noted in [83], but to incorporate these constraints in a way that allows for robust clustering.

SUPERPAC can be thought of as an extension of ideas from PCC literature [83, 74, 1] to leverage prior knowledge about the underlying geometry of the data. For datasets such as Yale B and MNIST, the strong subspace structure makes Euclidean distance a poor proxy for similarity between points in the same cluster, leading to the superior performance of our algorithm demonstrated in the following sections. This structure does not exist in all datasets, in which case we do not expect our algorithm to outperform current PCC algorithms. The reader will note we made a choice to order the certain sets according to the UoS model; this is similar to the choice in [1] to query according to similarity, where our notion of similarity here is based on subspace distances. We found this resulted in significant performance benefits, matching our intuition that points are clustered based on their nearest subspace. In contrast to [74, 1], where the test point is chosen according to a global improvement metric, we choose test points according to their classification margin. In our experiments, we found subspace margin to be a strong indicator of which points are misclassified, meaning that our algorithm rapidly corrects the errors that occur as a result of unsupervised subspace clustering.

Finally, note that the use of certain sets relies on the assumption that the pairwise queries are answered correctly—an assumption that is common in the literature [83, 84, 1]. We also note that min-margin examples may also be difficult for a human labeler to distinguish. For example, in the Yale face database, many min-margin points correspond to images with significant shadow. An empirical study of human ability to provide correct pairwise constraints as a function of margin would be an interesting topic for further study. However, in [1], the authors demonstrate that an algorithm based on certain sets still yields significant improvements under a small error rate. The study of robustly incorporating noisy pairwise comparisons is an interesting topic for further study.

Algorithm 3.2 UoS-EXPLORE

Input: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$: data, K: number of subspaces, d: dimension of subspaces, A: affinity matrix, maxQueries: maximum number of pairwise comparisons
Estimate Labels: $\hat{C} \leftarrow$ SPECTRALCLUSTERING(A,K)
Calculate Margin: Calculate margin according to (3.1) and set $x_\vee$ as the point of maximum margin (most confident point)
Initialize Certain Sets: $Z_1 = x_\vee$, $\mathcal{Z} = \{Z_1\}$, numQueries $\leftarrow 0$, $n_c \leftarrow 1$
while $n_c < K$ **and** numQueries $<$ maxQueries **do**
  Obtain Test Point: Choose x_T as point of maximum margin such that $\hat{C}(x_T) \neq \hat{C}(x \in Z_k)$ for any k. If no such x_T exists, choose x_T at random.
  Assign x_T to Certain Set:

    Sort $\{Z_1, \cdots, Z_{n_c}\}$ in order of most likely must-link (via subspace residual for x_T), query x_T against representatives from Z_k until must-link constraint is found or $k = n_c$. If no must-link constraint found, set $\mathcal{Z} \leftarrow \{Z_1, \cdots, Z_{n_c}, \{x_T\}\}$ and increment n_c.
end while

3.3.3 Initialization of Certain Sets

We now describe the process of initializing the certain sets. Note that this step is not necessary, as we could initialize all certain sets to be empty, but we found it led to improved performance experimentally. A main distinction between subspace clustering and the general clustering problem is that in the UoS model points can lie arbitrarily far from each other but still be on or near the same subspace. For this reason, the EXPLORE algorithm from [83] is unlikely to quickly find points from different clusters in an efficient manner. Here we define an analogous algorithm for the UoS case, termed UoS-EXPLORE, with pseudocode given in Algorithm 3.2. The goal of UoS-EXPLORE is to find K certain sets, each containing as few points as possible (ideally a single point), allowing us to more rapidly assign test points to certain sets in Algorithm 3.1. We begin by selecting our test point x_T as the most certain point, or the point of *maximum* margin and placing it in its own certain set. We then iteratively select x_T as the point of maximum margin that (1) is not in any certain set and (2) has a different cluster estimate from all points in the certain sets. If no such point exists, we choose uniformly at random from all points not in any certain set. This point is queried against a single representative from each certain set according to the UoS model as above until either a must-link is found or all set representatives have been queried, in which case x_T is added to a new certain set. This process is repeated until either K certain sets have been created or a terminal number of queries have been used. As points of maximum margin are more likely to be correctly clustered than other points in the set, we expect that by choosing points whose estimated labels indicate they do not belong to any current certain set, we will quickly find a point with no must-link constraints. We show in Section 3.4 that this algorithm finds at least one point from each cluster in nearly the lower limit of $K(K-1)/2$ queries on the Yale dataset.

3.4 Experimental Results

We compare the performance of our method and the nonparametric version of the URASC algorithm (URASC-N)[2] over a variety of datasets. Note that while numerous PCC algorithms exist, URASC achieves both the best empirical results and computational complexity on a variety of datasets. We also compared with the methods from [83] and [74] but found both to perform significantly worse than URASC on all datasets considered, with a far greater computational cost in the case of [74]. We use a maximum query budget of $2K$ for UoS-EXPLORE and EXPLORE. For completeness, we also compare to random constraints, in which queries are chosen uniformly at random from the set of unqueried pairs.

Finally, we compare against the oracle PCA classifier, which we now define. Let U_k be the d-dimensional PCA estimate of the points whose true label $C(x) = k$. Then the oracle label is $\hat{C}_o(x) = \arg\min_{k\in[K]} \left\|x - U_k U_k^T x\right\|_2$. This allows us to quantitatively capture the idea that, because the true classes are not perfectly low-rank, some points would not be clustered with the low-rank approximation of their own true cluster. In our experiments, we also compared with oracle robust PCA [93] implemented via the augmented Lagrange multiplier method [94] but did not find any improvement in classification error.

3.4.1 Error Metric

Many error metrics are considered throughout both the subspace clustering and general clustering literature. To allow for the most natural comparison with existing subspace clustering literature, we compare the clustering error, which is computed by matching the true labels and the labels output by a given clustering algorithm,

$$\text{err} = 100\left(1 - \max_{\pi} \frac{1}{N}\sum_{i,j} Q^{\text{out}}_{\pi(i)j} Q^{\text{true}}_{ij}\right),$$

where π is a permutation of the cluster labels, and Q^{out} and Q^{true} are the output and ground-truth labelings of the data, respectively, where the (i,j)th entry is one if point j belongs to cluster i and is zero otherwise.

Dataset	N	K	D	d
Yale	320-2432	5,10,38	2016	9
MNIST	500-1000	5,10	784	3
COIL-20	1440	20	1024	9
COIL-100	7200	100	1024	9
USPS	9298	10	256	15

Table 3.1: Datasets used for experiments with relevant parameters; N: total number of samples, K: number of clusters, D: ambient dimension, d: estimated subspace dimension.

3.4.2 Datasets

We consider five datasets commonly used as benchmarks in the subspace clustering literature[3], with a summary of the datasets and their relevant parameters are given in Table 3.1. The Yale B dataset consists of 64 images of size 192×168 of each of 38 different subjects under a variety of lighting conditions. For values of K less than 38, we follow the methodology of [95] and perform clustering on 100 randomly selected subsets of size K. We choose $d = 9$ as is common in the literature [23, 41]. The MNIST handwritten digit database test dataset consists of 10,000 centered 28×28 pixel images of handwritten digits 0-9. We follow a similar methodology to the previous section and select 100 random subsets of size K, using subspace dimension $d = 3$ as in [41]. The COIL-20 dataset [10] consists of 72 images of size 32×32 of each of 20 objects. The COIL-100 dataset [11] contains 100 objects (distinct from the COIL-20 objects) of the same size and with the same number of images of each object. For both datasets, we use subspace dimension $d = 9$. Finally, we apply our algorithm to the USPS dataset provided by [9], which contains 9,298 *total* images of handwritten digits 0-9 of size 16×16 with roughly even label distribution. We again use subspace dimension $d = 9$.

3.4.3 Input Subspace Clustering Algorithms

A major strength of our algorithm is that it is agnostic to the initial subspace clustering algorithm used to generate the input affinity matrix. To demonstrate this fact, we apply our algorithm with an input affinity matrix obtained from a variety of subspace clustering methods, summarized in Table 3.1. Note that some recent algorithms are not included in the simulations here. However, the simulations show that our algorithm works well with *any* initial clustering, and hence we expect similar results as new algorithms are developed.

[2]In our experiments, the parametric version of URASC was found to be numerically unstable and did not have significantly different performance from URASC-N in the best cases.

[3]The validity of the UoS assumption for two of these datasets is investigated in [23, 41].

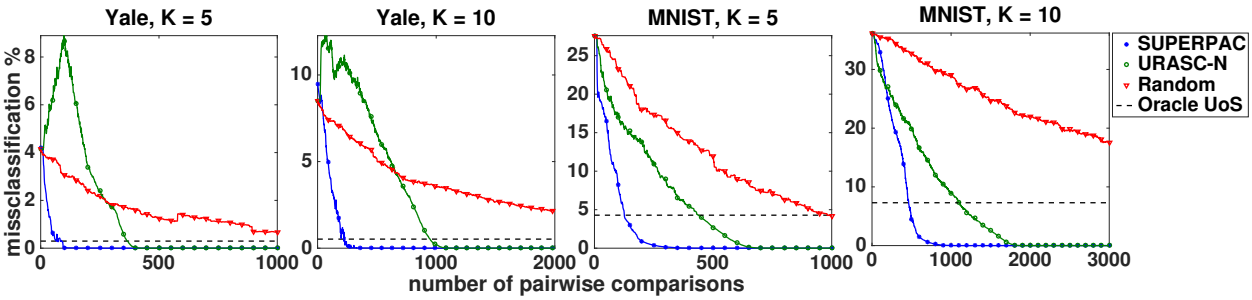

Figure 3.4: Misclassification rate for Yale B and MNIST datasets with many pairwise comparisons. Left-to-right: Yale B $K = 5$ (input from SSC), Yale B $K = 10$ (input from SSC), MNIST $K = 5$ (input from TSC), MNIST $K = 10$ (input from TSC).

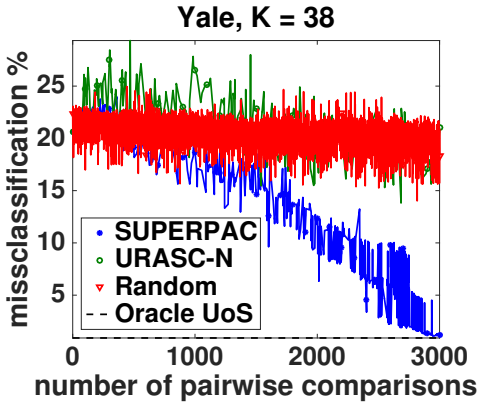

Figure 3.5: Misclassification rate versus number of pairwise comparisons for extended Yale face database B with $K = 38$ subjects. Input affinity matrix is taken from SSC-OMP.

3.4.4 Experimental Results

Fig. 3.4 shows the clustering error versus the number of pairwise comparisons for the Yale and MNIST datasets. The input affinity matrix is obtained by running SSC for the Yale dataset and by running TSC for the MNIST dataset. The figure clearly demonstrates the benefits of leveraging UoS structure in constrained clustering—in all cases, SUPERPAC requires roughly *half* the number of queries needed by URASC to achieve perfect clustering. For the Yale dataset with $K = 5$, roughly $2Kd$ queries are required to surpass oracle performance, and for $K = 10$ roughly $3Kd$ queries are required. Note that for the Yale dataset, the clustering error *increases* using URASC. This is due to the previously mentioned fact that imputing the wrong constraints can lead to worse clustering performance. For sufficiently many queries, the error decreases as expected. Fig. 3.5 shows the misclassification rate versus number of points for all $K = 38$ subjects of the Yale database, with the input affinity matrix taken from SSC-OMP [76]. We space out the markers for clearer plots. In this case, URASC performs roughly the same as random query selection, while SUPERPAC performs significantly better.

K	2	5	7	10
UoS-Explore	**1 (1/1)**	**10 (10/10)**	**21.58 (21/21)**	**48.6 (45/68)**
Explore [83]	4.57 (1/23)	117.68 (11/217)	259.93 (22/449)	494.8 (86/646)
Lower Bound	1	10	21	45

Table 3.2: Average number of queries to initialize K certain sets on Yale B dataset with 5th/95th quantiles given in parentheses. Smallest in bold.

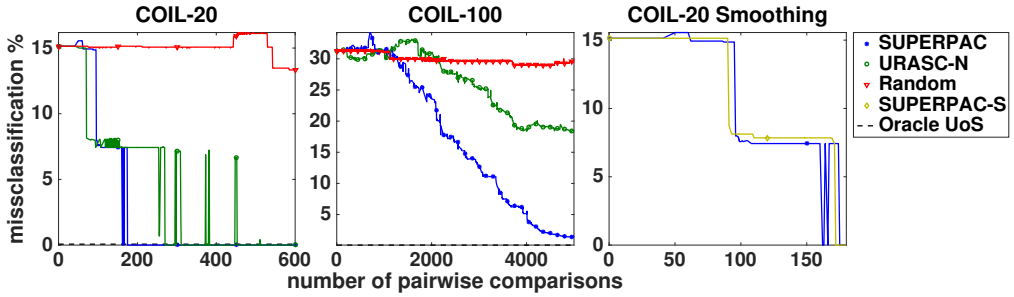

Figure 3.6: Misclassification rate versus number of pairwise comparisons for COIL-20 ($K = 20$) and COIL-100 ($K = 100$)) databases. Input affinity matrix is taken from EnSC. Rightmost plot shows proposed smoothing heuristic.

Next, we show the effectiveness of the UoS-EXPLORE algorithm over EXPLORE used in [83, 1]. We run the algorithms on 100 random subsets of K faces and report the average number of queries required to obtain K certain sets in Table 3.2. The table shows that UoS-EXPLORE uses far fewer queries to obtain K unique certain sets, with residual-based margin using very near the minimum required $K(K-1)/2$ queries. Note that in SUPERPAC and URASC, the query budget for this initialization step is limited in practice, and hence our method is more likely to discover K disjoint certain sets.

Fig. 3.6 demonstrates the continued superiority of our algorithm in the case where UoS structure exists. In the case of COIL-20, the clustering is sometimes unstable, alternating between roughly 0% and 7% clustering error for both active algorithms. This further demonstrates the observed phenomenon that spectral clustering is sensitive to small perturbations. To avoid this issue, we kept track of the K-subspaces cost function (see [13]) and ensured the cost decreased at every iteration. We refer to this added heuristic as SUPERPAC-S in the figure. The incorporation of this heuristic into our algorithm is a topic for further study.

Fig. 3.7 shows the resulting error on the USPS dataset, again indicating the superiority of our method. Note that N is large for this dataset, making spectral clustering computationally burdensome. Further, the computational complexity of URASC is dependent on N. As a result, URASC did not complete 2000 queries in 48 hours of run time when using 10 cores, so we compare to the result after completing only 1000 queries. Finally, in Fig. 3.8, we demonstrate that even on data without natural subspace structure, SUPERPAC performs competitively with URASC.

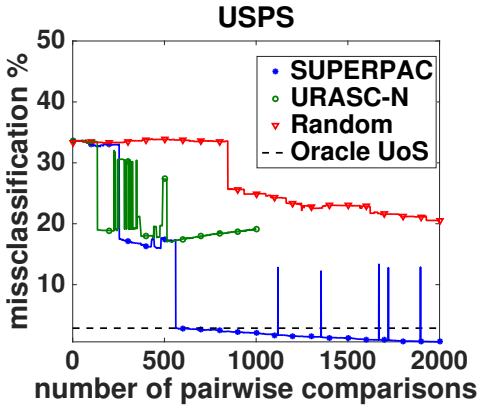

Figure 3.7: Misclassification rate versus number of pairwise comparisons for USPS dataset with $K = 10$ digits, 9,298 total samples. Input affinity matrix is taken from EnSC. URASC did not complete after 48 hours of run time.

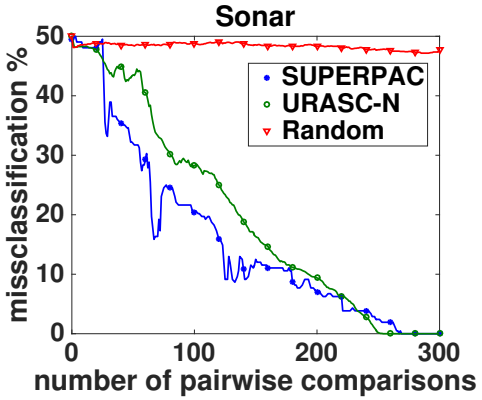

Figure 3.8: Misclassification rate for Sonar dataset from [1], where there is not reason to believe the clusters have subspace structure. We are still very competitive with state-of-the-art.

3.4.5 Computation Time

We compare the methods of query selection in terms of computational time on several datasets. Random querying can be selected offline and requires negligible computational time. Table 3.3 shows the average time per query for each of the three query selection methods along with the 5th and 95th quantiles in parentheses. The table clearly demonstrates the dependence on N of URASC and D for SUPERPAC. In the Yale dataset, where D is large relative to N and K, we see that URASC achieves significantly faster clustering. However, as both N and K increase, the SUPERPAC methods are faster by anywhere from one to three orders of magnitude, making our method extremely competitive from a computational perspective.

Algorithm	Yale, $K = 5$ $N = 320$ $D = 2016$, $d = 9$	Yale, $K = 10$ $N = 640$ $D = 2016$, $d = 9$	Yale, $K = 38$ $N = 2432$ $D = 2016$, $d = 9$	COIL, $K = 20$ $N = 1440$ $D = 1024$, $d = 9$	COIL, $K = 100$ $N = 7200$ $D = 1024$, $d = 9$	USPS, $K = 10$ $N = 9298$ $D = 256$, $d = 15$
SUPERPAC	1.40 (1.38/1.43)	2.78 (2.76/2.79)	10.42 (9.57/10.98)	0.44 (0.37/0.48)	5.78 (5.53/6.02)	0.19 (0.17/0.20)
URASC-N	0.11 (0.08/0.13)	0.28 (0.23/0.40)	6.38 (5.35/7.22)	4.61 (2.58/5.55)	252.97 (110.63/356.49)	155.02 (53.19/190.86)

Table 3.3: Average computation time (in seconds) per query required by PCC query selection algorithms on real datasets with 5th/95th quantiles given in parentheses.

3.5 Conclusion

We have presented a method of selecting and incorporating pairwise constraints into subspace clustering that considers the underlying geometric structure of the problem. The union of subspaces model is often used in computer vision applications where it is possible to request input from human labelers in the form of pairwise constraints. We showed that labeling is often necessary for subspace classifiers to achieve a clustering error near zero; additionally, these constraints can be chosen intelligently to improve the clustering procedure overall and allow for perfect clustering with a modest number of requests for human input.

We see this work as a bridge between adaptive query selection for pairwise constrained clustering and adaptive sampling for structured signals. (e.g., see previous work on sparse [89, 90], structured sparse [91], and low rank signals [92]). Several works apply ideas of compressive sensing to clustering [96, 97] and classification [88, 98], and recent work in subspace clustering has also shown that it's possible to cluster columns that lie in a union of subspaces even using compressed or subsampled data [99, 100, 101, 102]. A key interesting open question is whether the number of active queries required for clustering increases as the compression of the data increases. To address this we first plan to develop techniques for handling noisy query responses. In the case of face images, one may assume that compressed data would be harder to distinguish, leading to noisier query responses.

Another important topic for future study is that of convergence rates for our proposed SUPERPAC algorithm. The difficulty for this analysis lies in predicting the clustering output from spectral clustering; hence, the use of a different clustering method such as single linkage may provide an avenue for further analysis. One major drawback to our proposed algorithm is that it requires running spectral clustering after every test point is assigned to a certain set—a process that is known to be computationally burdensome. One possible means of overcoming this issue would be to select several test points at each round. As points of minimum margin may all lie in one subspace, it may be advantageous to explore other methods of choosing test points that balance uncertainty with diversity. Alternatively, since we are only changing one column and row of the affinity matrix with each test point, it may be appropriate to incrementally update the clustering, which would dramatically reduce the computational cost of spectral clustering.

We also saw that for datasets with different types of cluster structure, the structure assumptions

of each algorithm had direct impact on performance; in the future we plan to additionally develop techniques for learning from unlabeled data whether the union of subspace model or a standard clustering approach is more appropriate. Finally, we developed a notion of margin here for the purpose of selecting test points. It would be interesting to study whether subspace margin can be used in the context of supervised classification to train max-margin classifiers in a manner analogous to support vector machines.

CHAPTER 4

Ensemble Methods for Subspace Clustering

4.1 Introduction

The work of this chapter was performed jointly with David Hong and Dejiao Zhang. Specifically, Lemma 4.1 was formulated with significant support from David Hong and Dejiao Zhang. A first proof of the lemma was provided by Dejiao Zhang, and the current simplified form was completed by David Hong.

In modern computer vision problems such as facial recognition [7] and object tracking [103], researchers have found success applying the union of subspaces (UoS) model, in which data vectors lie near one of several subspaces. Under this model, the goal is to simultaneously identify these underlying subspaces and cluster the points according to their nearest subspace. Algorithms designed to solve this problem fall under the category of *subspace clustering*, a topic that has received a great deal of attention in recent years [73] due to its efficacy on real-world datasets such as the Extended Yale Face Database B [104] and the MNIST handwritten digit database [8].

One of the earliest approaches to solving the subspace clustering problem involves an iterative method in the spirit of K-means, known as K-subspaces (KSS) [13, 14, 15], which alternates between assigning points to clusters and estimating the subspace basis associated with each cluster. As this algorithm is only guaranteed to converge to a local minimum, in practice one runs many instances of the algorithm and chooses the final clustering as the one that produces the minimum cost. Although its empirical performance is limited, KSS continues to serve as a benchmark for subspace clustering algorithms, in part due to its computational efficiency and simplicity. Therefore, a deeper understanding of this method is an important contribution in the area of subspace clustering and a contribution of this chapter.

While the KSS cost function and alternating algorithm are perhaps the most natural approach for the subspace clustering problem, it is known that there is a set of initializations of nonzero measure from which the algorithm will convergence to a point other than the global minimizer.[1] Our

[1] We prove this fact for the simple case of two one-dimensional subspaces in $\mathbb{R}^2$ in Appendix C.

key observation is that even those "bad" initializations very commonly give some partially-correct clustering behavior and may be combined to form a more accurate clustering algorithm.

Our contributions are as follows. We introduce a novel application of the well-known *evidence accumulation clustering* framework [16] that leverages ensembles of the KSS algorithm to perform subspace clustering. By combining the results of many random initializations of KSS, this algorithm obtains an affinity matrix (known as a *co-association matrix*), to which we apply spectral clustering. We provide theoretical guarantees regarding the resulting affinity matrix that lead to recovery guarantees for the subspace clustering problem. We show that our method is extremely effective on both synthetic and real datasets; we show on synthetic data that our method has superior performance for subspaces that are extremely close together. Further, we show that a variant of our algorithm achieves state-of-the-art performance on several real datasets, including error on the COIL-20 image database and full Yale B database that are 24% and 54% better than state-of-the-art, respectively. Finally, since our method relies on multiple independent initializations, it is inherently parallelizable. To the best of our knowledge, we provide the first theoretical guarantees characterizing the co-association matrix resulting from evidence accumulation, as well as the first recovery guarantees for any variant of the KSS algorithm.

4.2 Problem Formulation & Related Work

Consider a collection of points $\mathcal{X} = \{x_1, \ldots, x_N\}$ in $\mathbb{R}^D$ belonging to a union of K subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_K$ having dimensions $d_1, \ldots, d_K$. Let $X \in \mathbb{R}^{D \times N}$ denote the matrix whose columns are the elements of $\mathcal{X}$. The goal of subspace clustering is to label points in the unknown union of K subspaces according to their nearest subspace. Once the clusters have been obtained, the corresponding subspace bases can be recovered using principal components analysis (PCA).

Most state-of-the-art approaches to subspace clustering rely on a *self-expressiveness* property of the data, which informally states that points in the UoS model can be most efficiently represented by other points within the same subspace. These methods typically use a self-expressive data cost function that is regularized to enforce efficient representation as follows:

$$\begin{aligned} \min_{Z} \quad & \|X - XZ\|_F^2 + \|Z\| \\ \text{subject to} \quad & \operatorname{diag}(Z) = 0, \end{aligned}$$

where $\|Z\|$ may be the 1-norm as in sparse subspace clustering (SSC) [23], nuclear norm as in low-rank representation (LRR) [78], or a combination of these and other norms. An affinity/similarity matrix is then obtained as $|Z| + |Z|^T$, after which spectral clustering is performed. Other terms are considered in the optimization problem to provide robustness to noise and outliers, and numerous

recent papers follow this framework [43, 105, 106, 107]. For large datasets, solving the above problem may be prohibitive, and algorithms such as [76, 77] employ orthogonal matching pursuit and elastic-net to provide reduced computational complexity and improved connectivity. Other approaches include thresholded subspace clustering (TSC) [41], in which an affinity matrix is formed by finding nearest neighbors of points in terms of spherical distance, and greedy subspace clustering (GSC) [40], which greedily builds subspaces in order to form an affinity matrix. In all cases, spectral clustering is performed as the final step to obtain cluster labels. One drawback to the above methods is that they exhibit poor performance when the subspaces of interest are close in terms of their principal angles. Under this setting, all points can be efficiently expressed by all other points in the dataset, presenting a challenge for regression-based methods. Similarly, points from any pair of subspaces may have large inner product, resulting in failure for TSC and GSC.

In contrast to the above methods, KSS seeks to minimize the sum of residuals of points to their assigned subspace, i.e.,

$$\min_{\mathcal{C},\mathcal{U}} \sum_{k=1}^{K} \sum_{i:x_i \in c_k} \left\| x_i - U_k U_k^T x_i \right\|_2^2, \tag{4.1}$$

where $\mathcal{C} = \{c_1, \ldots, c_K\}$ denotes the set of estimated clusters and $\mathcal{U} = \{U_1, \ldots, U_K\}$ denotes the corresponding set of subspace bases. Beginning with an initialization of K candidate subspace bases, KSS proceeds in an alternating fashion by (i) clustering points via nearest subspace and (ii) obtaining new subspace bases by performing PCA on the points in each cluster. The algorithm is computationally efficient and guaranteed to converge to a local minimum [13, 14]. As with K-means, the KSS output is highly dependent on initialization. It is typically applied by performing many restarts and choosing the result with minimum cost (4.1) as the output. This idea was extended to minimize the ℓ_1 norm in [108], where a method for intelligent initialization is also proposed. In [109], the authors use an alternating method based on KSS to perform online subspace clustering in the case of missing data. Most recently, in [110], the authors propose a novel initialization method based on ideas from [95], and then perform the subspace update step using gradient steps along the Grassmann manifold. While this method is computationally efficient and improves upon the previous performance of KSS, it lacks theoretical guarantees.

Initialization is important for KSS because it is known that there is a set of initializations of nonzero measure such that the algorithm will necessarily converge to a collection of subspaces and a clustering that do not globally minimize the cost in Eq. (4.1).[2] Our key observation is that even those "bad" initializations very commonly give some partially-correct clustering behavior and may be combined to form a more accurate clustering algorithm.

Ensemble methods have been used in the context of general clustering for some time and

[2]We prove this fact for the simple case of two one-dimensional subspaces in $\mathbb{R}^2$ in Appendix C.

with notable benefits being improved clustering performance as well as the ability to evaluate the performance of any individual clustering of the data. Such methods fall within the domain of *consensus clustering*, with an overview of the benefits and techniques given in [111]. The central idea behind these methods is to obtain many clusterings from a simple base clusterer, such as K-means, and then combine the results intelligently. In order to obtain different clustering results from each base clustering, diversity of some sort must be incorporated. This is typically done by obtaining bootstrap samples of the data as in [112, 113], subsampling the data to reduce computational complexity as in [114], or performing random projections of the data [115]. Alternatively, the authors of [116, 117] use the randomness in different initializations of K-means to obtain diversity, which is the approach we take here for subspace clustering. After diversity is achieved, the base clustering results must be combined. The *evidence accumulation clustering* framework is laid out in [16], in which results are combined by voting, i.e., creating a co-association matrix A whose (i, j)th entry is equal to the number of times two points are clustered together. A theoretical framework for this approach is laid out in [118], where the entries of the co-association matrix are modeled as Binomial random variables. This approach is studied further in and a soft clustering is obtained via a matrix factorization formulation. This idea is extended in the work of [119, 120], in which the clustering problem is solved as a Bregman divergence minimization. These models result in improved clustering performance over previous work but are not accompanied by any theoretical guarantees with regard to the resulting co-association matrix. Further, they are not specifically designed to consider the case where the data are generated from the UoS model.

In the remainder of this chapter, we apply ideas from consensus clustering to the subspace clustering problem. We describe our ensemble KSS algorithm and its guarantees and demonstrate the algorithm's state-of-the-art performance on synthetic and several real datasets.

4.3 Ensemble K-Subspaces Algorithm & Guarantees

In this section, we describe our method for subspace clustering using ensembles of the K-subspaces algorithm, which we refer to as Ensemble K-subspaces (EKSS).

EKSS leverages the fact that for several runs of KSS, each random initialization results in some partially correct clustering information. We therefore run several random initializations of KSS and form a co-association matrix using the results of each run, after which we apply spectral clustering. Our theoretical results imply that even if the data come from generative subspaces with arbitrary positioning, the algorithm outputs a perfect clustering, as long as the maximum subspace affinity (defined below in Eq. (4.2)) is bounded and the points are drawn uniformly from the true subspaces without noise. For noisy data, the final affinity matrix contains no false connections between points.

In more technical detail, our EKSS algorithm proceeds as follows. For each of $b = 1, \ldots, B$ base

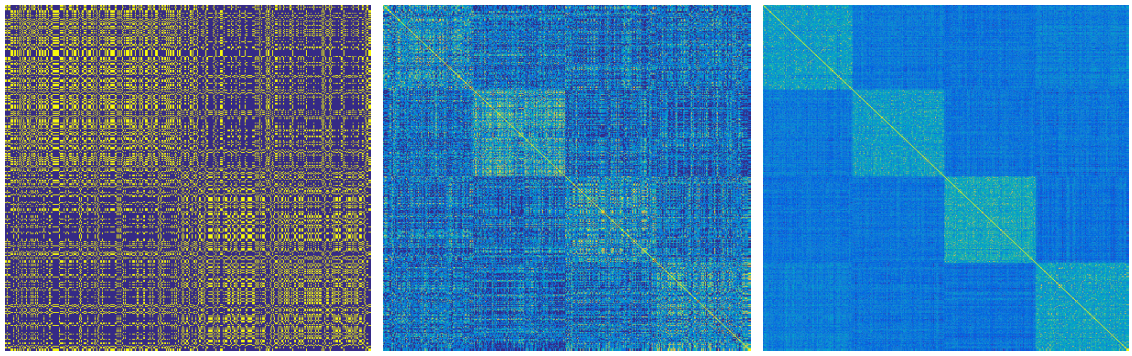

Figure 4.1: Co-association matrix of EKSS for $B = 1, 5, 50$ base clusterings. Data generation parameters are $D = 100$, $d = 10$, $K = 4$, $N = 400$, and the data is noise-free; the algorithm uses $\bar{K} = 4$ candidate subspaces of dimension $\bar{d} = 10$. Resulting clustering errors are 54%, 25%, and 0%.

clusterings, we obtain a cluster estimate $\mathcal{C}^{(b)}$ from a single run of KSS with a random initialization. For each b such that the points x_i and x_j are clustered together, we add a fixed constant to the (i, j)th entry of the co-association matrix. We then threshold the co-association matrix as in [41] by taking the top q values from each row/column. Once this thresholded co-association matrix is formed, cluster labels are obtained using spectral clustering. Pseudocode for EKSS is given in Alg. 4.1, where THRESH sets all but the top q entries in each row/column to zero as in [41] (pseudocode for this procedure is given in Alg. 4.3) and SPECTRALCLUSTERING [121] clusters the data points based on the co-association matrix A. Note that the number of candidates $\bar{K}$ and candidate dimension $\bar{d}$ need not match the number K and dimension of the true underlying subspaces. Fig. 4.1 shows the progression of the co-association matrix as $B = 1, 5, 50$ base clusterings are used, in the case of noiseless data from $K = 4$ subspaces of dimension $d = 10$ in ambient space of dimension $D = 100$ using $\bar{K} = 4$ candidates of dimension $\bar{d} = 10$.

While the final clustering could be obtained using hierarchical methods as in [16], optimization techniques as in [119, 120], or another method of choice. In our experiments, we found spectral clustering to result in better clustering performance than either of the previously mentioned approaches, with a lower computational cost than the approach used in [120].

4.3.1 Recovery Guarantees

Recovery guarantees for KSS are still absent despite nearly twenty years of use since its introduction. Intelligent initialization methods based on probabilistic farthest insertion are provided in [108, 110], but these still lack any theoretical guarantees. In this section, we provide a first step toward understanding the performance of KSS, as well as recovery guarantees for the subspace clustering problem. We show that by combining the clusterings that result from many random initializations

Algorithm 4.1 ENSEMBLE K-SUBSPACES (EKSS)

Input: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^D$: data, $\bar{K}$: number of candidate subspaces, $\bar{d}$: candidate dimension, K: number of output clusters, q: threshold parameter, B: number of base clusterings, T: number of KSS iterations
Output: $\mathcal{C} = \{c_1, \ldots, c_K\}$: clusters of $\mathcal{X}$
for $b = 1, \ldots, B$ (in parallel) **do**
  $U_1, \ldots, U_{\bar{K}} \overset{iid}{\sim} \mathrm{Unif}(\mathrm{St}(D, \bar{d}))$ — Draw $\bar{K}$ random subspace bases
  $c_k \leftarrow \left\{ x \in \mathcal{X} \ : \ \forall j \ \left\| U_k^T x \right\|_2 \geq \left\| U_j^T x \right\|_2 \right\}$ for $k = 1, \ldots, \bar{K}$ — Cluster by projection
  for $t = 1, \ldots, T$ (in sequence) **do**
    $U_k \leftarrow \mathrm{PCA}\left(c_k, \bar{d}\right)$ for $k = 1, \ldots, \bar{K}$ — Estimate subspaces
    $c_k \leftarrow \left\{ x \in \mathcal{X} \ : \ \forall j \ \left\| U_k^T x \right\|_2 \geq \left\| U_j^T x \right\|_2 \right\}$ for $k = 1, \ldots, \bar{K}$ — Cluster by projection
  end for
  $\mathcal{C}^{(b)} \leftarrow \{c_1, \ldots, c_{\bar{K}}\}$
end for
$A_{i,j} \leftarrow \frac{1}{B} \left| \left\{ b : x_i, x_j \text{ are co-clustered in } \mathcal{C}^{(b)} \right\} \right|$ for $i, j = 1, \ldots, N$ — Form affinity matrix
$\bar{A} \leftarrow \mathrm{THRESH}(A, q)$ — Keep top q entries per row/column
$\mathcal{C} \leftarrow \mathrm{SPECTRALCLUSTERING}(\bar{A}, K)$ — Final Clustering

of subspace candidates, the entries of the resulting affinity matrix converge to a monotonically increasing function of the absolute value of inner product between points. A corollary of this fact is that a simplified version of EKSS exhibits all the recovery guarantees of TSC [41]. To the best of our knowledge, our work is the first to provide any theoretical guarantees for the KSS algorithm as well as the first characterization of the co-association matrix in the context of consensus clustering.

Due to its alternating nature, analyzing multiple iterations of the KSS algorithm remains challenging. Instead, we analyze the first iteration of KSS ($T = 0$ in Alg. 4.1), in which random candidates are drawn and points are clustered based on their nearest candidate. Further, we restrict ourselves to the case where the number of candidates (*not* the number of subspaces) is $\bar{K} = 2$ and the candidate dimension (*not* the true subspace dimension) is $\bar{d} = 1$. Finally, for the purposes of analysis, we replace the unit norm candidates in Step 4 with Gaussian random vectors, noting that the two are nearly equivalent in high dimensions due to concentration of the norm [39, Thm. 3.1.1]. We refer to EKSS with this choice of parameters as EKSS-0 and include explicit pseudocode in 4.2. Remarkably, we show that combining the results from many random instances of this naïve algorithm leads to the same recovery guarantees as TSC, which are in turn comparable to those for SSC. While we do not analyze the case where multiple KSS iterations are performed, these iterations are guaranteed not to increase the KSS cost function, and in practice, we find that running KSS to convergence only improves clustering performance.

We now state our main result, which guarantees that EKSS-0 described in the preceding paragraph is able to cluster the points in $\mathcal{X}$ exactly under given conditions on the maximum affinity

Algorithm 4.2 EKSS-0

Input: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^D$: data, K: number of output clusters, $q \in \mathbb{N}$: threshold parameter, B: number of base clusterings
Output: $\mathcal{C} = \{c_1, \ldots, c_K\}$: clusters of $\mathcal{X}$
for $b = 1, \ldots, B$ (in parallel) **do**
  $u_1, u_2 \sim \mathcal{N}\left(0, \frac{1}{D} I_D\right)$ — Draw two random candidates
  $c_k \leftarrow \left\{x \in \mathcal{X} \ : \ \left|u_k^T x\right| \geq \left|u_l^T x\right|\right\}$ for $k, l = 1, 2$ — Cluster by inner product
  $\mathcal{C}^{(b)} \leftarrow \{c_1, c_2\}$
end for
$A_{i,j} \leftarrow \frac{1}{B}\left|\left\{b : x_i, x_j \text{ are clustered together in } \mathcal{C}^{(b)}\right\}\right|$ for $i, j = 1, \ldots, N$ — Form affinity matrix
$\bar{A} \leftarrow \textsc{Thresh}(A, q)$ — Threshold
$\mathcal{C} \leftarrow \textsc{SpectralClustering}(\bar{A}, K)$ — Final Clustering

between subspaces and the number of points per subspace. We note that this guarantee relies on knowledge of the true number of clusters K for the SPECTRALCLUSTERING step. Without such knowledge, the result reduced to the "no false connections" guarantee of Thm. 4.2. This result matches that of TSC [41, Thm. 2] exactly, and our proof leverages the proof of that used in [41] by applying Lemma 1 below. Our guarantees depend on the affinity between two subspaces, defined as [41, 122]

$$\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l) = \frac{1}{\sqrt{d_k \wedge d_l}} \left\|U_k^T U_l\right\|_F, \tag{4.2}$$

where $\mathcal{S}_k$ and $\mathcal{S}_l$ are d_k- and d_l-dimensional subspaces with orthonormal bases U_k and U_l. Note that $\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l)$ is a measure of how close two subspaces are in terms of their principal angles and takes the value 1 if two subspaces are equivalent and 0 if they are orthogonal.

Theorem 4.1. *Let $\mathcal{S}_k$, $k = 1, \ldots, K$ be subspaces of dimension $d_1, \ldots, d_K$ in $\mathbb{R}^D$. Let the points in $\mathcal{X}_k$ be a set of N_k points drawn uniformly from the unit sphere in subspace k, i.e., from the set $\{x \in \mathcal{S}_k : \|x\| = 1\}$. Let $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$ and $N = \sum_k N_k$. Let $q \in [c_1 \log N_{max}, N_{min}/6]$, where $N_{min} = \min_k \{N_k\}$, $N_{max} = \max_k \{N_k\}$, $c_1 = 18(12\pi)^{d-1}$, and $d = \max\{d_1, \ldots, d_K\}$. If*

$$\max_{k,l:k\neq l} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l) \leq \frac{1}{15 \log N},$$

then in the limit as $B \to \infty$, EKSS-0 delivers the correct clustering of $\mathcal{X}$ with probability at least $1 - 10/N - \sum_{k=1}^K \left(N_k e^{-c(N_k - 1)} + 2N_k^{-2}\right)$, where $c > 0$ is a numerical constant.

Proof of Theorem 4.1. The proof hinges on the following lemma, which states that in the case where $T = 0$, $\bar{d} = 1$, and $\bar{K} = 2$, points x_i and x_j are clustered together with probability that increases monotonically with $\left|x_i^T x_j\right|$.

Lemma 4.1. *The probability that two points $x_i, x_j \in \mathcal{X}$ are clustered together by one base clustering of EKSS-0 (i.e., EKSS-0 with $B = 1$) is an increasing function of $\left|x_i^T x_j\right|$.*

The proof of Lemma 4.1 is given in Appendix C. By the Law of Large Numbers, when $B \to \infty$, each entry $A_{i,j}$ of the affinity matrix A approaches the probability analyzed in Lemma 4.1, and hence is also an increasing function of $\left|x_i^T x_j\right|$. Next, note that the result of [41, Thm. 2] depends only on the relative order of $\left|x_i^T x_j\right|$ (namely, through [41, Lemma 1] and [41, Lemma 2]). By Lemma 4.1, the order of entries in A is the same as in TSC, and so (as $B \to \infty$) the thresholded affinity matrix $\bar{A}$ of EKSS-0 has the same connectivity as that formed by TSC [41]. The result of the theorem follows directly by the proof of [41, Thm. 2]. □

Thm. 4.1 states that perfect clustering of the data is guaranteed even in the case of intersecting subspaces, as long as the subspaces are not too close in all directions. The clustering condition for Thm. 4.1 above is the same as that for SSC in [43] up to constants and log factors. Along with the above result, all recovery guarantees of TSC follow from Lemma 4.1, indicating that EKSS-0 results in no false connections under noisy data, missing data, and outliers. We state the result for noisy data here for completeness.

Theorem 4.2. *Let the points in $\mathcal{X}_k$ be the set of N_k points $x_i^{(k)} = y_i^{(k)} + e_i^{(k)}$, where the $y_i^{(k)}$ are drawn i.i.d. from the set $\{x \in \mathcal{S}_k : \|x\| = 1\}$, independently across k, and the $e_i^{(k)}$ are i.i.d. $\mathcal{N}(0, \frac{\sigma^2}{D} I_D)$. Let $\mathcal{X} = \mathcal{X}_1 \cup \cdots \cup \mathcal{X}_K$ and $q \leq N_{min}/6$, where $N_{min} = \min_k \{N_k\}$. If*

$$\max_{k,l:k\neq l} \operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l) + \frac{\sigma(1+\sigma)}{\sqrt{\log N}} \frac{\sqrt{d}}{\sqrt{D}} \leq \frac{1}{15 \log N},$$

with $d > 6 \log N$, then in the limit as $B \to \infty$, $\bar{A}$ obtained from running EKSS-0 has no false connections with probability at least $1 - 10/N - \sum_{k=1}^{K} N_k e^{-c(N_k - 1)}$, where $c > 0$ is a numerical constant.

Proof. By Lemma 4.1 above, the order of the entries in A remains the same as in TSC, and hence the proof follows directly. □

For a discussion of these guarantees and their relation to those for SSC and other algorithms, see [41, Sec. VII]. The inverse dependence on $\log N$ implies that the subspace affinity must shrink as N grows. On one hand, this is intuitive because with many points per subspace, it is likely that some points will be arbitrarily close to the intersection of two subspaces and potentially be misclustered. On the other hand, more points allows for a better chance that the nearest point by inner product is within the same subspace. Indeed, in all the empirical results we see that both EKSS and TSC perform better with larger N. Finally, we note that while the above analysis holds only for the case of $T = 0$, letting $T > 0$ is guaranteed not to increase the KSS cost function [13]. The extension of Thm. 4.1 to the case where $T > 0$ is an important topic of our ongoing research.

Algorithm 4.3 AFFINITY THRESHOLD (THRESH)

Input: $A \in [0,1]^{N \times N}$: affinity matrix, q: threshold parameter
Output: $\bar{A} \in [0,1]^{N \times N}$: thresholded affinity matrix
for $i = 1, \ldots, N$ **do**
  $Z^{\text{row}}_{i,:} \leftarrow A_{i,:}$ with the smallest $N - q$ entries set to zero. Threshold rows
  $Z^{\text{col}}_{:,i} \leftarrow A_{:,i}$ with the smallest $N - q$ entries set to zero. Threshold columns
end for
$\bar{A} \leftarrow \frac{1}{2}\left(Z^{\text{row}} + Z^{\text{col}}\right)$ Average

4.3.2 Implementation Details

In this section, we explain a few relevant implementation details, including a warm start extension of EKSS that we will show outperforms state-of-the-art methods on several benchmark datasets.

4.3.2.1 Thresholding Procedure

The pseudocode for the thresholding procedure THRESH as given in Alg. 4.3, which results in the same connectivity as thresholding in TSC [41]. Recall that the purpose of this subroutine is to set all but the top q entries from each row/column of the affinity matrix to zero. Note that this procedure could be applied to *any* affinity matrix, such as those resulting from SSC or its variants. However, the theoretical motivation for doing so is unclear, and in our brief experiments, we did not find any significant benefit provided. The principled application of thresholding as a post-processing procedure is an interesting topic for future studied.

4.3.2.2 Base Clustering Accuracy

A natural heuristic to improve the clustering performance of EKSS is to add larger values to the affinity matrix for base clusterings in which the clustering is believed to be more accurate, and smaller values in the case where the clustering is believed to be more inaccurate. Here, we briefly describe one such approach. Note that Step 12 in EKSS is equivalent to setting

$$A \leftarrow \frac{1}{B} \sum_{b=1}^{B} A^{(b)} \phi(b),$$

where $A^{(b)}_{i,j} := \mathbf{1}_{x_i, x_j \text{ are clustered together in } \mathcal{C}^{(b)}}$ and $\phi(b) = 1$. One measure of clustering accuracy is the cost function of KSS with the clusters and subspace bases set as the clustering output. Let $\mathcal{C}^{(b)} = \left\{c^{(b)}_1, \ldots, c^{(b)}_K\right\}$, and let $\mathcal{U}^{(b)} = \left\{U^{(b)}_1, \ldots, U^{(b)}_K\right\}$ denote the set of subspace bases estimated by performing PCA on the points in the corresponding clusters. The clustering confidence can then

Algorithm 4.4 EKSS WARM-START (EKSS-WS)

Input: $\mathcal{X} = \{x_1, x_2, \ldots, x_N\} \subset \mathbb{R}^D$: data, $\bar{K}$: number of candidate subspaces, $\bar{d}$: candidate dimension, K: number of output clusters, q_1, q_2: threshold parameters, B_1, B_2: number of base clusterings, T: number of KSS iterations
Output: $\mathcal{C} = \{c_1, \ldots, c_K\}$: clusters of $\mathcal{X}$
for $b = 1, \ldots, B_1$ (in parallel) **do**
  $\{c_1, \ldots, c_{\bar{K}}\} \leftarrow \text{EKSS}\left(\mathcal{X}, \bar{K}, \bar{d}, \bar{K}, q_2, B_2, T\right)$
  for $t = 1, \ldots, T$ (in sequence) **do**
    $U_k \leftarrow \text{PCA}\left(c_k, \bar{d}\right)$ for $k = 1, \ldots, \bar{K}$ Estimate subspaces
    $c_k \leftarrow \left\{x \in \mathcal{X} \ : \ \forall j \ \left\|U_k^T x\right\|_2 \geq \left\|U_j^T x\right\|_2\right\}$ for $k = 1, \ldots, \bar{K}$ Cluster by projection
  end for
  $\mathcal{C}^{(b)} \leftarrow \{c_1, \ldots, c_{\bar{K}}\}$
end for
$A_{i,j} \leftarrow \frac{1}{B_1}\left|\left\{b : x_i, x_j \text{ are clustered together in } \mathcal{C}^{(b)}\right\}\right|$ for $i, j = 1, \ldots, N$ Form affinity matrix
$\bar{A} \leftarrow \text{THRESH}(A, q_1)$ Keep top q_1 entries per row/column
$\mathcal{C} \leftarrow \text{SPECTRALCLUSTERING}(\bar{A}, K)$ Final Clustering

be measured as

$$\phi(b) = 1 - \sum_{k=1}^{K} \sum_{i: x_i \in c_k^{(b)}} \left\| x_i - U_k^{(b)} U_k^{(b)^T} x_i \right\|_2^2 / \left\| X \right\|_F^2, \tag{4.3}$$

a value between 0 and 1 that decreases as the KSS cost increases. We employ this value of $\phi(b)$ in all experiments on real data.

4.3.2.3 Warm-Start EKSS

It is well-known that the performance of alternating methods in optimization depends on the initialization of the problem parameters [123, 124]. For this reason, we propose a warm-start method to further improve robustness to outliers and noise. We first run EKSS with a small number of base clusterings (typically 10). Then, using the estimated labels obtained from this run, we form a set of initial candidate subspace bases by performing PCA on the points in each cluster. These candidate bases are then used to initialize KSS in place of random candidates for each $b = 1, \ldots, B$ in EKSS. We refer to this variant as EKSS-WS (warm-start), and pseudocode for this algorithm is given in Alg. 4.4.

4.3.2.4 Parameter Selection

In all experiments using EKSS, we take $\bar{K} = K$ and choose $\bar{d}$ as the best approximation of the true subspace dimension. We assume in this work that a good approximating dimension for the underlying subspace is known, which is reasonable in several practical applications. For example,

images of a Lambertian object under varying illumination are known to lie near a subspace with $d = 9$ [7] and moving objects in video are known to lie near an affine subspace with $d = 3$ [125]. In the case of unknown subspace dimensions, one could use EKSS with increasing $\bar{d}$, or methods such as those proposed in [15] could be employed.

Rather than choosing T explicitly in Alg. 4.1, we run KSS to convergence. The EKSS algorithm then relies on the appropriate choice of the number of base clusterings B and the thresholding parameter q. In general, B may be chosen as large as computation time allows. In our experiments on real data, we choose $B = 1000$. The thresholding parameter q can be chosen according to data-driven techniques as in [126], or following the choice in [41]. In our experiments on real data, for each EKSS and TSC we try a large range of values q and select the q that achieves the lowest KSS cost 4.1. For the warm-start run of EKSS-WS, both B_2 and q_2 should be small (we choose $B_2 = 10$ and $q_2 = 3$). For comparison, TSC requires the choice of q, and SSC [23] and its variants [76, 77] all require two parameters to be selected. Finally, we use the implementation of SPECTRALCLUSTERING from [23].

As a final note, we also experimented with applying ideas of subsampling to SSC-OMP [76] and EnSC [77]. However, the resulting clustering performance did not always surpass that of the base algorithm run on the full dataset. The investigation of principled techniques for applying evidence accumulation methods to these algorithms is an interesting topic for future research.

4.4 Experimental Results

In this section, we demonstrate the performance in terms of clustering error (defined below) of EKSS on both synthetic and real datasets. We first show the performance of our algorithm as a function of the relevant problem parameters and verify that EKSS-0 exhibits the same empirical performance as TSC. We also show that EKSS can recover subspaces that either have large intersection or are extremely close. We then demonstrate on real datasets that EKSS not only improves over previous iterative methods, but that the warm-start variant of EKSS surpasses state-of-the-art results in many cases.

4.4.1 Error Metric

Many error metrics are considered throughout both the subspace clustering and general clustering literature. To allow for the most natural comparison with existing subspace clustering literature, we compare the clustering error, which is computed by matching the true labels and the labels output

by a given clustering algorithm,

$$\text{err} = 100\left(1 - \max_{\pi} \frac{1}{N} \sum_{i,j} Q^{\text{out}}_{\pi(i)j} Q^{\text{true}}_{ij}\right), \tag{4.4}$$

where π is a permutation of the cluster labels, and Q^{out} and Q^{true} are the output and ground-truth labelings of the data, respectively, where the (i, j)th entry is one if point j belongs to cluster i and is zero otherwise.

4.4.2 Synthetic Data

For all experiments in this section, we take $q = \max(3, \lceil N_k/20 \rceil)$ for EKSS-0 and TSC and $q = \max(3, \lceil N_k/6 \rceil)$ for EKSS, where $\lceil c \rceil$ denotes the largest integer greater than or equal to c. We set $B = 10,000$ for EKSS-0 and EKSS. To validate our theoretical results, we draw Gaussian candidates, rather than orthonormal bases, for EKSS-0. When the angles between subspaces are not explicitly specified, it is assumed that the subspaces are drawn uniformly at random from the set of all d-dimensional subspaces of $\mathbb{R}^D$. For all experiments, we draw points uniformly at random from the unit sphere in the corresponding subspace and show the mean error over 100 random problem instances. We use the code provided by the authors for TSC and SSC. We employ the ADMM implementation of SSC and choose the parameters that result in the best performance in each scenario.

4.4.2.1 Verification of Theoretical Results

We first verify the results of Section 4.3.1 through simulation. We demonstrate the dependence of co-clustering on inner product, where we say that points x_i and x_j are co-clustered if both points have maximum inner product with the same candidate basis. Fig. 4.2 shows the empirical probability of co-clustering as a function of the absolute inner product between points for the case of $D = 100$, $d = 10$, where we range over 1000 values of the inner product and take 10,000 random instances for each value. As expected, two points are co-clustered with frequency monotonically increasing with their inner product. Further, we see that using d-dimensional subspace bases only improves co-clustering. Extending Lemma 4.1 to this case analytically is an important piece of future work.

4.4.2.2 Influence of Problem Parameters

Having verified the theoretical results of Section 4.3.1, we now explore the influence of the relevant problem parameters on the EKSS algorithm.

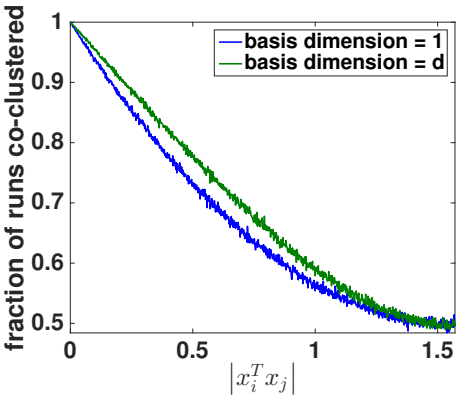

Figure 4.2: Empirical probability of co-clustering as a function of the inner product between points.

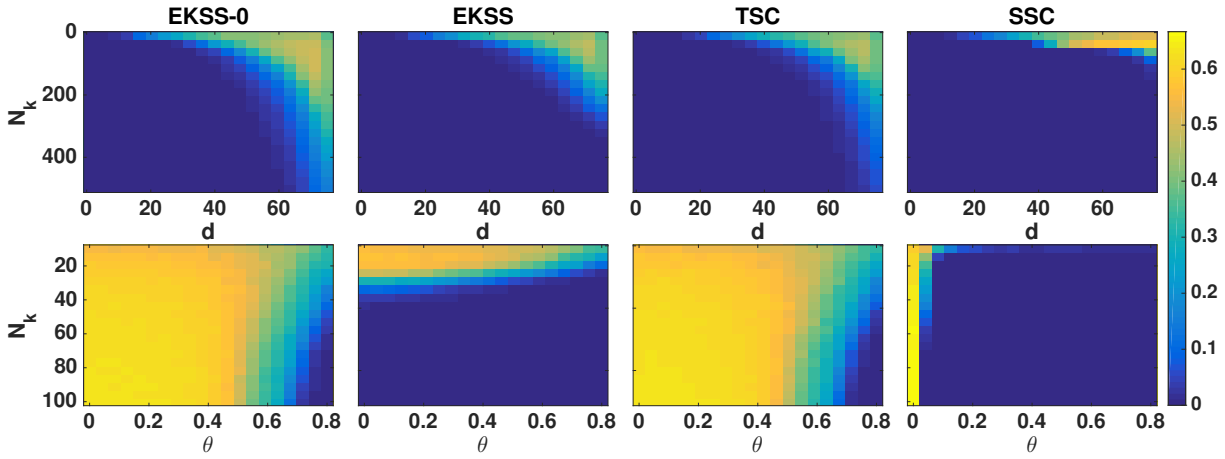

Figure 4.3: Clustering error for proposed and state-of-the-art subspace clustering algorithms as a function of problem parameters N_k, number of points per subspace, and true subspace dimension d or angle between subspaces θ. Fixed problem parameters are $D = 100$, $K = 3$.

We explore the influence of some relevant problem parameters on the EKSS algorithm in Fig. 4.3. We take the ambient dimension to be $D = 100$, the number of subspaces to be $K = 3$, and assume the data to be noiseless.

We first explore the dependence on subspace dimension and the number of points per subspace. The top row of Fig. 4.3 shows the misclassification rate as the number of points per subspace ranges from $10 - 500$ and the subspace dimension ranges from $1 - 75$. When $2d > D$ ($d = 51$ in this case), pairs of subspaces necessarily have intersection, and the intersection dimension grows with d. First, the figures demonstrate that EKSS-0 achieves roughly the same performance as TSC, resulting in correct clustering even in the case of subspaces with large intersection. Second, we see that EKSS can correctly cluster for subspace dimensions larger than that of TSC as long as there are sufficiently many points per subspace. For large subspace dimensions with a moderate number of points per subspace, SSC achieves the best performance.

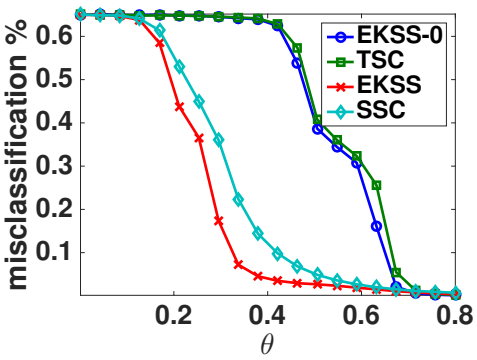

Figure 4.4: Clustering error as a function of subspace angles with noisy data. Problem parameters are $D = 100$, $d = 10$, $K = 3$, $N_k = 500$, $\sigma^2 = 0.05$.

We next explore the clustering performance as a function of the distance between subspaces, as shown in the second row of Fig. 4.3. We set the subspace dimension to $d = 10$ and generate $K = 3$ subspaces such that all principal angles are θ, for 20 values in the range $[0.001, 0.8]$. Most strikingly, EKSS is able to resolve subspaces with even the smallest separation. This stands in contrast to TSC, which fails in this regime because when the subspaces are extremely close, the inner products between points on different subspaces can be nearly as large as those within the same subspace. Similarly, in the case of SSC, points on a different subspace can be used to regress any given point without any added cost, and so it fails at very small subspace angles. However, as long as there is still some separation between subspaces, EKSS is able to correctly cluster all points. While the theory presented here does not capture this phenomenon, recovery guarantees that take into account multiple iterations of KSS are an important topic for future work.

We also consider the effect of additive noise on clustering performance by adding Gaussian noise with zero mean and covariance $\sigma^2 I_D$. The third row of Fig. 4.3 shows the misclassification rate as a function of points per subspace and noise variance. Under this setting, all three algorithms achieve roughly the same performance, with EKSS resulting in the lowest classification error.

As a final comparison, we show the clustering performance with noisy data. Fig. 4.4 shows the clustering error as a function of the angle between subspaces for the case of $K = 3$ subspaces of dimension $d = 10$, with $N_k = 500$ points corrupted by zero-mean Gaussian noise with covariance $0.05 I_D$. The figure shows again that EKSS-0 and TSC obtain similar performance, and more importantly that EKSS is more robust to small subspace angles than SSC, even in the case of noisy data.

4.4.3 Real Data

In this section, we show that EKSS achieves competitive performance on a variety of real datasets commonly used as benchmarks in the subspace clustering literature. The comparison presented

Dataset	N	K	D	d
Hopkins-155	39-556	2-3	30-200	3
Yale	2432	38	2016	9
COIL-20	1440	20	1024	9
COIL-100	7200	100	1024	9
USPS	9298	10	256	15
MNIST-10k	10000	10	500	3

Table 4.1: Datasets used for experiments with relevant parameters; N: total number of samples, K: number of clusters, D: ambient dimension, d: estimated subspace dimension.

here is among the best-known *subspace clustering algorithms*, and hence better unsupervised performance may be achieved by algorithms relying on a different clustering model.

4.4.3.1 Details and Preprocessing of Datasets

In this section, we describe the real datasets used in our experiments, as well as any preprocessing steps and the parameters selected for all algorithms. All datasets are normalized so that each column lies on the unit sphere in the corresponding ambient dimension, as is common in the literature [43, 41, 110]. Table 4.1 gives a summary of all datasets considered.

The Hopkins-155 dataset [103] consists of 155 motion sequences with $K = 2$ in 120 of sequences and $K = 3$ in the remaining 35. In each sequence, objects moving along different trajectories each lie near their own affine subspace of dimension at most 3. We perform no preprocessing steps on this dataset and report both the mean and median misclassification rates, as is common in the literature [23, 110].

The Extended Yale Face Database B [104, 127] consists of 64 images of each of 38 different subjects under a variety of lighting conditions. Each image is of nominal size 192×168 and is known to lie near a 9-dimensional subspace [7]. We downsample so that each image is of size 48×42, as in [23]. For EKSS, EKSS-WS, KSS, MKF, TSC, and OLRSC, we perform an initial whitening as in [95, 41] by removing the first two singular components of the dataset and then project the data onto its first 500 principal components to reduce the computational complexity of these methods. Whitening resulted in worse performance for all other algorithms, so we omitted this step.

The COIL-20 [10] and COIL-100 [11] datasets consist of 72 images of 20 and 100 distinct objects (respectively) under a variety of rotations. All images are of size 32×32. On both datasets, we whiten by removing the first singular component when it improves algorithm performance.

The USPS dataset provided by [9] contains 9,298 total handwritten digits of size 16×16 with roughly even label distribution. No preprocessing is performed on this dataset.

The MNIST dataset [8] contains a total of 70,000 handwritten digits, of which we consider

Algorithm	Hopkins	Yale	COIL-20	COIL-100	USPS	MNIST-10k
EKSS	$B = 1000, q = 15$	$B = 1000, q = 5$	$B = 1000, q = 35$	$B = 1000, q = 11$	$B = 1000, q = 7$	$B = 1000, q = 9$
EKSS-WS	$B_1 = 1000, q_1 = 15$ $B_2 = 10, q_2 = 3$	$B_1 = 1000, q_1 = 19$ $B_2 = 10, q_2 = 3$	$B_1 = 1000, q_1 = 50$ $B_2 = 10, q_2 = 3$	$B_1 = 1000, q_1 = 60$ $B_2 = 10, q_2 = 3$	$B_1 = 1000, q_1 = 500$ $B_2 = 10, q_2 = 3$	$B_1 = 1000, q_1 = 8$ $B_2 = 10, q_2 = 3$
TSC	$q = 3$	$q = 3$	$q = 8$	$q = 8$	$q = 5$	$q = 9$
SSC-ADMM	$\rho = 0.7, \alpha = 800$	$\rho = 1, \alpha = 20$	$\rho = 1, \alpha = 20$	$\rho = 1, \alpha = 20$	$\rho = 1, \alpha = 20$	$\rho = 1, \alpha = 20$
SSC-OMP	$\varepsilon = 2^{-52}, k_{max} = 3$	$\varepsilon = 2^{-52}, k_{max} = 5$	$\varepsilon = 2^{-52}, k_{max} = 5$	$\varepsilon = 2^{-52}, k_{max} = 5$	$\varepsilon = 2^{-52}, k_{max} = 5$	$\varepsilon = 2^{-52}, k_{max} = 12$
EnSC	$\lambda = 0.1, \alpha = 3$	$\lambda = 0.95, \alpha = 3$	$\lambda = 0.95, \alpha = 3$	$\lambda = 0.95, \alpha = 3$	$\lambda = 0.95, \alpha = 50$	$\lambda = 0.95, \alpha = 3$
OLRSC	F	S	F	F	F	F

Table 4.2: Parameters used in experiments on real datasets for all algorithms considered.

only the 10,000 "test" images. The images have nominal size 28×28, and we use the output of a scattering convolutional network [128] of size 3,472 and then project onto the first 500 principal components as in [77].

4.4.3.2 Algorithms for Comparison

We compare the performance of EKSS to several benchmark algorithms: KSS [13], Median K-Flats (MKF) [108], TSC [41], SSC-OMP [76], Elastic Net Subspace Clustering (EnSC) [77], and Online Low-Rank Subspace Clustering (OLRSC) [107]. For EKSS we use $B = 1000$ base clusterings. For all others but KSS, we use the code provided by the authors and use the recommended parameters where available and otherwise use parameters that result in the best performance. For a fair comparison to KSS and MKF, we run 1000 trials of each and use the clustering result that achieves the lowest KSS cost. We refer to the warm-start variant of EKSS as EKSS-WS.

The parameters used for all experiments are shown in Table 4.2. We use the recommended parameters where available and choose the parameters that result in the best performance in all other cases. For OLRSC [107], we use the recommended parameters, set the basis dimension as $K * d$, and report the minimum error between the standard and fully online pipelines, indicated by 'S' and 'F' in the table.

4.4.3.3 Results

The clustering error for all datasets and algorithms is shown in Table 4.3, with the lowest two errors given in bold. First, note that EKSS outperforms KSS in all cases, and typically by a very large margin. This result emphasizes the importance of leveraging all clustering information from the B base clusterings, as opposed to simply choosing the best single clustering. Next, the results show that EKSS-WS is among the top two performers in all datasets considered. Although code for the method from [110] was unavailable, EKSS-WS achieves similar performance to the reported misclassification rate on the Hopkins-155 dataset. We also observe that scalable algorithms such as SSC-OMP and EnSC perform poorly on the Hopkins dataset, likely due to the small number of points, whereas EKSS works well under both small and large N. Most striking are the resulting misclassification rates on the Yale B and COIL-20 datasets, for which EKSS-WS significantly

Algorithm	Hopkins	Yale B	COIL-20	COIL-100	USPS	MNIST-10k
EKSS	5.84/0.32	22.33	24.79	36.36	33.21	26.18
EKSS-WS	**3.82/0.00**	**16.12**	**7.01**	**28.03**	**26.09**	**17.54**
KSS	7.00/0.79	75.70	65.56	74.53	51.30	48.15
MKF	5.22/0.21	47.70	54.79	66.49	**28.62**	47.14
TSC	27.19/27.27	20.81	15.35	39.03	33.46	**17.17**
SSC-ADMM	**2.18/0.00**	31.03	22.43	44.06	56.61	19.17
SSC-OMP	35.77/36.65	22.41	46.67	62.12	78.43	19.47
EnSC	24.96/24.21	**21.26**	**15.14**	**28.75**	33.66	17.97
OLRSC	20.70/18.12	55.14	35.42	50.79	29.71	20.50

Table 4.3: Clustering error of subspace clustering algorithms for a variety of benchmark datasets. Hopkins-155 performance is (mean/median). The lowest two clustering errors are given in bold.

outperforms the best existing algorithm. Interestingly, TSC achieves the best performance on the MNIST-10k dataset, whereas EnSC achieves much better performance on the full 70,000-digit database as reported in [77]. Due to memory constraints we were unable to compare performance on the full MNIST database for most algorithms including EKSS. Implementing a memory-efficient version of EKSS is an important topic of future work.

4.5 Conclusion

In this work, we presented a first step toward a theoretical understanding of the KSS algorithm by analyzing the effect of combining multiple clusterings using the evidence accumulation clustering framework. We showed that with a given choice of parameters, our algorithm can provably cluster data from a union of subspaces under the same conditions as existing algorithms. We demonstrated the efficacy of our approach on both synthetic and real data, and showed that a warm-start variant of our method achieves excellent performance on several real datasets.

While the theoretical guarantees presented here match existing guarantees in the literature, our experiments on synthetic data indicate that the iterative approach of KSS provides a major improvement in robustness to small angles between subspaces. Extending our analysis to multiple iterations of KSS would perhaps provide stronger theoretical guarantees that illuminate this relaxed assumption on the subspace affinity. A full convergence analysis of KSS would also be of general interest to the subspace clustering community. Further, while our results hold only for the case of two 1-dimensional candidates, we observed a performance improvement in the case where more than two d-dimensional candidates are used. Extending our analysis to the general case of Alg. 4.1 (e.g., $T > 0$, $\bar{d} > 1$, and $\bar{K} > 2$) is an important next step that is difficult since our analysis currently relies heavily on a characterization of the Gaussianity of the inner products that no longer holds in the case of higher dimensional candidates. Another interesting avenue for future

exploration would be to apply ensemble-type methods to other base clustering algorithms. We briefly experimented with ensembles of self-expressive methods, obtaining diversity in each base clustering by subsampling the data. However, our simulations found the clustering performance to be unpredictable, sometimes dramatically increasing the resulting clustering error. In Section 4.3.2, we discussed a number of heuristics to improve the performance of EKSS. One approach that was not discussed would be to use some robust subspace estimation techniques in place of PCA within the KSS algorithm (line 7, Alg. 4.1). The very recent algorithm of [129] exhibits both strong robustness to outliers and low computational complexity, making it a strong candidate for such a procedure. While EKSS-WS with the given parameter choices achieves excellent empirical performance, a deeper understanding of this method could lead to improved performance and robustness across different datasets. In particular, it will be important to study why the parameters chosen exhibit such strong performance. Finally, EKSS-WS can be viewed as a sort of network architecture with only two layers. It would be interesting to determine whether there exist other, more principled, architectures that lead to improved performance, though adding further layers would result in an increased computational cost.

CHAPTER 5

Clustering Quality Measures for Subspace Clustering

5.1 Introduction

The problem of subspace clustering has gained traction in recent years due to its excellent empirical performance on computer vision problems including facial recognition [104, 127] and object tracking [103], as well as algorithmic theoretical guarantees characterizing when correct clustering can be achieved. While existing algorithms such as Elastic Net Subspace Clustering (EnSC) [77] and Ensemble K-subspaces ([24], Chapter 4) are both scalable and principled, these and other algorithms require the selection of a number of tuning parameters. For example, all methods relying on the *self-expressive* property of data from a union of subspaces require tuning at least one hyperparameter that balances the regression term with other norm penalties. Geometric methods such as K-subspaces [13, 14, 15] and Greedy Subspace Clustering (GSC) [40] rely on some knowledge of the underlying subspace dimensions, and Thresholded Subspace Clustering (TSC) [41] and EKSS [24] require selecting the thresholding parameter. While nearly all subspace clustering methods rely on some form of parameter selection, to the best of our knowledge, existing work does not consider the selection of such parameters in a principled manner. In this chapter, we aim to solve this problem by studying *clustering quality measures* that are specific to the union-of-subspaces (UoS) model.

Methods to evaluate clustering instances in the absence of ground truth have been considered in previous work under the names *clustering quality measures* (CQMs) [25] and *internal clustering validation measures* [130, 131]. In contrast to the supervised learning setting, clustering problems do not provide any labeled data that can be used as a "hold-out" set for cross-validation. For this reason, researchers in this field attempt to design quality metrics that provide a measure of confidence in a given clustering. Such measures are designed to capture the "natural" goals of clustering, the chief being that points within clusters should have high similarity relative to points across different clusters. However, as we will show through our experimental results, existing CQMs do not provide reliable confidence measures for subspace clustering problems, and hence the

development of specialized CQMs for this context is necessary for the advance of this field. To the best of our knowledge, no study of CQMs exists for the specific case where the data of interest lie on a union of subspaces.

Unlike the general clustering problem, subspace clustering assumes a geometric model for the data with a natural objective. Therefore, many of the ambiguities that accompany the general clustering problem may be avoided, and the idea of one (or few) CQMs applying to all subspace clustering algorithms is reasonable. In [132], the authors argue that lack of interpretability plagues modern clustering algorithms and accounts for the widespread use of K-means in spite of its known shortcomings. Subspace clustering falls victim to a similar problem, as relatively few people understand the concept of a union of subspaces, perhaps accounting for its relative anonymity among practitioners.[1] For this paradigm to gain popularity, the ability to select parameters is paramount, and hence the need to compare clusterings resulting from different subspace clustering algorithms is an important contribution that has received no attention to this point.

In this chapter, we study the problem of clustering quality measures specifically designed for the subspace clustering problem. We present three CQMs for UoS data and demonstrate their efficacy in choosing both the dimensions of the underlying subspaces and the appropriate parameters for a wide variety of subspace clustering algorithms. We show through simulations on synthetic and real data that these outperform existing CQMs in terms of selecting the parameters that correspond to the lowest clustering error. Finally, we discuss the axiomatic study of clustering quality and develop analogs of existing axioms that are amenable to data lying on a union of subspaces.

5.2 Related Work

As mentioned in the previous section, existing CQMs aim to capture basic properties of clusterings such as similarity within and between clusters. Surveys of existing CQMs are given in [130, 131], and CQMs such as the Dunn index [21], Silhouette index [22], and Davies-Bouldin index [133] are widely used for comparing clusterings when a notion of distance between points exists. A more recent line of work [25, 134, 135, 136] attempts to form an axiomatic framework describing properties any reasonable CQM should satisfy; we defer the discussion of such ideas to Section 5.5 and restrict ourselves to empirically-driven CQMs in this section. We briefly describe three popular CQMs here to provide insight into existing methods. The Dunn index is the ratio of inter-cluster similarity to intra-cluster similarity, where the former is defined as the minimum distance between points in different clusters and the latter is the maximum distance among all pairs of points in the same cluster. The Silhouette index relies on the difference between the similarity of a point

[1] For example, there is not a single subspace clustering algorithm implemented in the widely-used scikit-learn Python package.

to its own cluster and its next most similar cluster. The Davies-Bouldin index considers both within-cluster scatter, measured as the distances of points to their nearest cluster centroid, and the similarities between cluster centroids.

In general, these and other measures penalize clusterings whose inter-cluster pairwise distances are similar to their intra-cluster pairwise distances. For points lying on a subspace, pairwise distance is not indicative. For example, the points x and $-x$ clearly lie on the same one-dimensional subspace but may be arbitrarily far apart. Therefore, neither the Dunn index nor Silhouette index has a natural analog for UoS data. The Davies-Bouldin index can be modified by measuring within-cluster similarity as the distance from a point to its nearest subspace, and cluster-cluster similarity through the principal angles between subspaces. However, in our experiments, we did not find this CQM to reliably select clusterings with low error.

The work mentioned thus far focuses on the general case, where the only requirement is a distance function between points. However, many modern clustering algorithms rely only on the entries of an adjacency matrix, whose (i, j)th entry $A_{ij} \in \{0, 1\}$ denotes whether two items in the set are "connected," or an affinity matrix, whose entries $A_{ij} \geq 0$ denote the strength of that connection. Such algorithms are referred to as *graph-based* methods and include single linkage, other hierarchical methods, and spectral clustering (see [137, Ch. 14] for a description of these methods).[2] Empirical graph clustering quality measures have existed for a number of years, and several empirical comparisons of such metrics exist [138, 139, 140], with no CQM consistently outperforming others when a large number of datasets are considered. Two of the most widely-used CQMs are *coverage* [141] and *modularity* [142]. The former is defined as the ratio of intra-cluster connectivity and total connectivity in the graph, and the latter measures the strength of intra-cluster connectivity compared to the average connectivity of each cluster (we define these CQMs formally in Section 5.4). While these and other CQMs perform reliably on a variety of datasets, they suffer from known drawbacks such as favoring sparse affinity matrices.

State-of-the-art methods in subspace clustering typically proceed by forming an affinity matrix and then performing spectral clustering to obtain label estimates. With this in mind, a natural course of action would be to apply existing graph-based CQMs to this affinity matrix. However, as mentioned in the previous section, the subspace clustering problem assumes a strong geometric model that should be taken into account when evaluating clustering quality. As with the case of active label requests in Chapter 3, we expect that by leveraging the underlying geometric structure, it should be possible to overcome existing limitations and develop CQMs that more accurately measure clustering quality in the case where the data lie on a union of subspaces.

[2]Note that these graphs are *often* derived from distances between points, but this is not always the case.

5.3 Quality Measures for Subspace Clustering

In this section, we propose three quality measures specifically designed for data lying on a union of subspaces. Consider a collection of points $\mathcal{X} = \{x_1, \ldots, x_N\}$ in $\mathbb{R}^D$, and let $X \in \mathbb{R}^{D \times N}$ denote the matrix whose columns are the elements of $\mathcal{X}$. We define a K-*clustering* of $\mathcal{X}$ to be a partition of $\mathcal{X}$ into K disjoint sets $\mathcal{C} = \{c_1, \cdots, c_K\}$, where we assume $1 < K < N$ to avoid trivial clustering. We assume that the data lie near a union of K subspaces $\mathcal{S}_1, \ldots, \mathcal{S}_K$ with corresponding dimensions $d_1, \ldots, d_K$. Under this model, we wish to label points in the unknown union of K subspaces according to their nearest subspace. Once the clusters have been obtained, the corresponding subspace bases are recovered using principal components analysis (PCA).

A major obstacle toward measuring quality of subspace clustering is that existing CQMs depend on some notion of distance between points. As mentioned above, for points lying on a subspace, pairwise distance is not indicative. For this reason, rather than considering distances between points, we base our CQMs on distances from points to subspaces. Let $U \in \mathbb{R}^{D \times d}$ be an orthonormal basis for a d-dimensional subspace $\mathcal{S}$. Then the distance from a point x to subspace $\mathcal{S}$ is defined as

$$\operatorname{dist}(x, \mathcal{S}) = \left\| x - UU^T x \right\|_2 . \tag{5.1}$$

Based on the above definition, we now define a CQM for UoS data as a function $m : \mathcal{C} \times \mathcal{X} \times \mathcal{U} \to \mathbb{R}_{\geq 0}$, where the subspace bases $\mathcal{U} = \{U_1, \cdots, U_K\}$ are those derived by performing PCA or some other subspace estimation technique on the data assigned to each cluster. For notational convenience, we formulate these such that smaller values indicate a better clustering. The first of these measures is that of the KSS cost function, which we refer to as m_{KSS}. Given a clustering $\mathcal{C} = \{c_1, \ldots, c_K\}$ and corresponding set of subspace bases $\mathcal{U} = \{U_1, \ldots, U_K\}$, we define the *KSS cost CQM* as

$$m_{\text{KSS}}(\mathcal{C}, \mathcal{X}, \mathcal{U}) = \frac{1}{N} \sum_{k=1}^{K} \sum_{i : x_i \in c_k} \operatorname{dist}(x_i, \mathcal{S}_k)^2 = \frac{1}{N} \sum_{k=1}^{K} \sum_{i : x_i \in c_k} \left\| x_i - U_k U_k^T x_i \right\|^2 . \tag{5.2}$$

Another possible CQM for the UoS model hinges on the idea of subspace margin presented in Chapter 3. Let $k(i)$ denote the cluster label for point x_i, and let

$$k'(i) = \arg\min_{j \neq k(i)} \operatorname{dist}(x_i, \mathcal{S}_j).$$

The *subspace margin CQM* is then defined as

$$m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \mathcal{U}) = \frac{1}{N} \sum_{i=1}^{N} \frac{\operatorname{dist}(x_i, \mathcal{S}_{k(i)})}{\operatorname{dist}(x_i, \mathcal{S}_{k'(i)})}. \tag{5.3}$$

Note that the subspace margin CQM is the average subspace margin of all points in the dataset and is similar to the Relative Margin CQM proposed in [25]. As with the KSS cost, smaller values of subspace margin indicate a better clustering quality.

While the above two CQMs are designed to capture the goodness of fit of the estimated subspaces to the data, they fall short in two key areas. First, neither method seems appropriate for selecting the number of clusters in the dataset, since both achieve a perfect fit (value of zero) for the trivial clustering of one point per cluster. Second, the CQMs are not amenable to selecting the estimated subspace dimension d, which is an input to the KSS and EKSS algorithms. Note that the KSS cost will generally decrease monotonically with d, since it is likely that the data vectors will have some noise and therefore span the entire space. The dependence of subspace margin on d is unclear and an important topic for future study. While these may seem to be major drawbacks, nearly all existing subspace clustering algorithms rely on spectral clustering and hence require the number of clusters to be known beforehand. In order to appropriately choose the subspace dimension, we propose a modified version of subspace margin that penalizes clusterings whose corresponding estimated subspaces are close. This CQM relies on the affinity between two subspaces, defined as

$$\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l) = \frac{1}{\sqrt{d_k \wedge d_l}} \left\| U_k^T U_l \right\|_F,$$

where $\mathcal{S}_k$ and $\mathcal{S}_l$ are d_k- and d_l-dimensional subspaces with orthonormal bases U_k and U_l. Note that $\operatorname{aff}(\mathcal{S}_k, \mathcal{S}_l)$ is a measure of how close two subspaces are in terms of their principal angles and takes the value 1 if two subspaces are equivalent and 0 if they are orthogonal. Let the average pairwise affinities between all subspaces be defined as

$$\overline{\operatorname{aff}} = \frac{2}{K(K-1)} \sum_{j \neq k} \operatorname{aff}(\mathcal{S}_j, \mathcal{S}_k),$$

where the implicit input is the set of K subspace bases. With this notion of subspace similarity in mind, we introduce the *normalized margin CQM*, which is defined as

$$m_{\text{norm}}(\mathcal{C}, \mathcal{X}, \mathcal{U}) = \frac{1}{N(1 - \overline{\operatorname{aff}})} \sum_{i=1}^{N} \frac{\operatorname{dist}(x_i, \mathcal{S}_{k(i)})}{\operatorname{dist}(x_i, \mathcal{S}_{k'(i)})}. \tag{5.4}$$

Consider two extreme examples of the above CQM. In the case where all subspaces are orthogonal, $\overline{\operatorname{aff}} = 0$, and $m_{\text{norm}}(\mathcal{C}, \mathcal{X}, \mathcal{U}) = m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \mathcal{U})$. However, when many subspaces are close in terms of their principal angles, $\overline{\operatorname{aff}}$ is large, inducing a heavy penalty for the clustering $\mathcal{C}$. As the subspace dimension increases, the subspaces are increasingly likely to have overlap in some directions, making their affinity larger. Therefore, we expect the normalized margin CQM to be appropriate for selecting the subspace dimension. We confirm this intuition through simulation in

Algorithm	Parameter 1	Description	Parameter 2	Description
SSC-ADMM	$\rho \in [0.1, 10]$	thresholding parameter	$\alpha \in [5, 2000]$	hyperparameter
SSC-OMP	$k_{max} \in [1, 50]$	maximum coefficients	-	-
EnSC	$\lambda \in [0.01, 0.99]$	hyperparameter	$\alpha \in [3, 100]$	hyperparameter
GSC	$k_{max} \in [1, 20]$	# neighbors	-	-
TSC	$q \in [1, N]$	thresholding parameter	-	-
EKSS	$q \in [1, N]$	thresholding parameter	$d \in [1, 25]$	subspace dimension

Table 5.1: Ranges of tuning parameters considered for various subspace clustering algorithms.

the next section.

5.4 Empirical Results

In this section, we demonstrate the utility of the proposed CQMs through experimental results on both synthetic and real datasets. We consider six state-of-the-art subspace clustering algorithms: the ADMM implementation of SSC (SSC-ADMM) [23], the solution to SSC via orthogonal matching pursuit (SSC-OMP) [76], Elastic Net Subspace Clustering (EnSC) [77], Greedy Subspace Clustering (GSC) [40], Thresholded Subspace Clustering (TSC) [41], and Ensemble K-subspaces (EKSS) [24]. These methods rely on widely different properties of UoS data and a variety of tuning parameters, including hyperparameters for optimization programs and thresholding parameters. Hence, we hope that any CQM that performs well across all methods will generalize well to any subspace clustering algorithm that exists. We run each algorithm over a wide range of tuning parameters shown in Table 5.1 and record the best clustering error[3] and the clustering error corresponding to the best clustering quality for each CQM.

We compare the performance of our proposed CQMs with two graph-based measures, coverage and modularity. Consider an affinity matrix A and clustering $\mathcal{C}$, and let $w_A = \sum_{i,j=1}^{N} A_{ij}$ be the sum of all weights in the graph. The *coverage* of the graph and clustering is defined as

$$\text{coverage}(\mathcal{C}, A) = \frac{\sum_{k=1}^{K} \sum_{i,j \in c_k} A_{ij}}{w_A}.$$

The *modularity* is defined as

$$\text{modularity}(\mathcal{C}, A) = \sum_{k=1}^{K} \left(\frac{\sum_{i,j \in c_k} A_{ij}}{w_A} - \left(\frac{\sum_{i \in c_k, j=1,\ldots,N} A_{ij}}{w_A} \right)^2 \right).$$

[3]Clustering error is defined as the number of misclassified points divided by the total number of points under the best permutation of labels. For a formal definition, see Eq. (4.4), Chapter 4.

	Best UoS	KSS Cost	Subspace Margin	Normalized Margin	Coverage	Modularity
Clustering Error	4.65	63.15	61.60	5.43	21.72	18.35
Subspace Dimension (mean/mode)	3/3	25/25	24.9/25	2.9/3	1.4/1	1.6/2
Clustering Error	0	7.73	1.13	0	0.03	0.03
Subspace Dimension (mean/mode)	10/10	21.6/22	19.7/20	10/10	9.2/9	9.2/9

Table 5.2: Performance of CQMs for selecting subspace dimension in EKSS algorithm on noisy UoS data from $K = 6$ random subspaces in $\mathbb{R}^{100}$ with $N_k = 100$ points drawn per subspace and noise variance $\sigma^2 = 0.05$. Top two rows: subspace dimension $d = 3$. Bottom two rows: subspace dimension $d = 10$. "Best UoS" indicates error when the true subspace dimension is given to EKSS.

5.4.1 Synthetic Data

We first consider the problem of selecting the appropriate subspace dimension via CQMs for the EKSS algorithm. We let the ambient dimension $D = 100$, true subspace dimension $d = 3$ and $d = 10$, number of subspaces $K = 6$, and points per subspace $N_k = 100$. We draw the subspaces at random and draw the data uniformly at random from the unit sphere intersected with the corresponding subspace. We then corrupt the points with isotropic, independent, additive Gaussian noise with variance $\sigma^2 = 0.05$. We generate ten random instances of the data and run EKSS with estimated subspace dimension ranging from 1-25. Table 5.2 shows the resulting error and subspace dimension corresponding to the best clustering chosen by each CQM. The results indicate that normalized margin is the clear choice for subspace dimension selection among all CQMs considered, selecting the true subspace dimension in nearly every instance. We also see that coverage and modularity choose clusterings with lower corresponding errors than the proposed CQMs but significantly underestimate the subspace dimension. Finally, both the KSS cost and subspace margin are biased toward large subspace dimensions, making them unfit CQMs for subspace dimension selection.

While the normalized margin CQM succeeds in selecting the correct underlying subspace dimension, our further experiments showed that it does not perform as well as KSS cost or subspace margin in the case where the subspace dimensions are known. For example, in the next scenario, the selected error was uniformly worse than that chosen by KSS cost or subspace margin by a range of $1 - 3\%$. Therefore, we do not include it in any of the remaining experiments of this section.

We now compare the CQMs on synthetic data with known subspace dimension under two scenarios known to be challenging for subspace clustering algorithms with the wrong parameter selection. In both scenarios, we let the ambient dimension $D = 100$, subspace dimension $d = 10$, number of subspaces $K = 10$, and points per subspace $N_k = 100$. We generate ten random instances of the data and report the average values for each algorithm and CQM.

In the first scenario, we draw the subspaces at random and draw the data uniformly at random from the unit sphere intersected with the corresponding subspace. We then corrupt the points with isotropic, independent, additive Gaussian noise with variance $\sigma^2 = 0.05$. Table 5.3 shows the

Algorithm	Minimum Error	KSS Cost	Subspace Margin	Coverage	Modularity
SSC-ADMM	2.66	2.79	2.90	70.50	69.95
SSC-OMP	7.78	7.78	7.78	66.72	66.72
EnSC	1.39	1.41	1.41	2.04	2.02
GSC	3.19	3.19	3.19	9.05	3.20
TSC	1.22	1.39	1.39	1.53	1.50
EKSS	1.09	1.13	1.17	2.19	1.16

Table 5.3: Performance of CQMs on noisy UoS data from $K = 10$ random subspaces of dimension $d = 5$ in $\mathbb{R}^{100}$ with $N_k = 100$ points drawn per subspace and noise variance $\sigma^2 = 0.05$.

average resulting errors selected by each CQM for each algorithm considered, where "Minimum Error" is the error corresponding to the best parameter selection among all tried (which often requires knowledge of the true labels for selection). The table shows that both proposed CQMs outperform existing graph-based CQMs in every case, typically selecting clusterings that achieve near-minimum error rates. Interestingly, coverage and modularity choose especially bad clusterings for SSC-ADMM and SSC-OMP. Examining the affinity matrices for the corresponding parameters, we see that the best clustering (highest coverage and modularity) typically corresponds to one of extremely high sparsity. This is a known feature of these CQMs [138] and should be accounted for when considering these methods. Finally, we see that modularity outperforms coverage for all algorithms.

In the second scenario, we consider the case where the subspaces are close together in terms of principal angles. We generate five subspaces at random, and for each of these five generate a second subspace whose principal angles are all fixed to $\theta = 0.1$. This setting is known to be challenging for all subspace clustering algorithms, as points from nearby subspaces have large inner product and high similarity, confounding both geometric and self-expressive methods. However, since the data are noiseless, for most algorithms the correct selection of parameters results in excellent clustering performance. We report the clustering performance corresponding to the various CQMs in Table 5.4. Again we see that the subspace-based CQMs consistently choose clusterings of minimum error, while the graph-based methods fail for all but the case of GSC. While these simulations are hardly extensive, this initial study indicates that when the data truly lie on a union of subspaces, the proposed CQMs significantly outperform existing popular methods.

An interesting question is that of sensitivity to the chosen parameters for each algorithm. However, as the subspace clustering algorithms considered rely on widely different techniques, only algorithm-specific comments can be made. In general, the algorithms are less sensitive to hyperparameters (e.g., in the case of SSC-ADMM and EnSC) and more sensitive to thresholding parameters (e.g., in the case of TSC and EKSS). This is an important point that we believe should be discussed thoroughly as new subspace clustering algorithms are proposed.

Algorithm	Minimum Error	KSS Cost	Subspace Margin	Coverage	Modularity
SSC-ADMM	0	0	0	53.79	21.27
SSC-OMP	0	0	0	22.37	0
EnSC	12.20	12.20	12.21	50.83	41.61
GSC	0	0	0	0	0
TSC	46.41	48.38	52.03	52.03	52.03
EKSS	0	0	0	39.37	1.24

Table 5.4: Performance of CQMs on noise-free UoS data from $K = 10$ random subspaces of dimension $d = 5$ in $\mathbb{R}^{100}$ with $N_k = 100$ points drawn per subspace. Subspaces are paired such that each has fixed principal angles $\theta = 0.1$ to one other subspace.

Dataset	N	K	D	d
Hopkins-155	39-556	2-3	30-200	3
Yale	2432	38	2016	9
COIL-20	1440	20	1024	9

Table 5.5: Real datasets used for experiments with relevant parameters; N: total number of samples, K: number of clusters, D: ambient dimension, d: estimated subspace dimension.

5.4.2 Real Data

The results from the previous section indicate that the proposed CQMs perform well under data that are truly generated from a union of subspaces. In this section, we examine the performance of all CQMs on three real benchmark datasets common to the subspace clustering literature. We consider the Hopkins-155 [103] dataset of motion sequences, the Extended Yale Face Database B [104, 127] of faces under varying illuminations, and the COIL-20 [10] dataset of images under a variety of rotations. Table 5.5 shows the datasets with their corresponding problem parameters. We preprocessed the data by removing the top principal components as in Chapter 4 for the Yale and COIL-20 datasets. Table 5.6 shows the corresponding best errors for each algorithm selected by each CQM. For the Hopkins dataset, we do not consider the results of SSC-OMP or EnSC, as these are known to perform poorly regardless of parameter choice. We likewise do not consider SSC-ADMM for the Yale dataset, as its computational complexity prohibits it from being run over a wide array of parameter choices.

In general, the table indicates that the KSS cost performs best across the largest number of datasets and algorithms. The Hopkins dataset is known to exhibit extremely strong UoS structure, a fact that is confirmed by the superior performance of both the KSS cost and subspace margin CQMs. In most cases, margin slightly outperforms KSS cost on this dataset. The results on the Yale dataset demonstrate the shortfalls of subspace margin, which selects very poor clusterings for all algorithms considered. This is likely due to the fact that the subspaces in this database have small principal angles and consist of noisy data. While subspace margin may be a strong choice for small angles in

Algorithm	Minimum Error	KSS Cost	Subspace Margin	Coverage	Modularity
Hopkins					
SSC-ADMM	3.60	5.93	5.60	19.33	24.04
SSC-OMP	-	-	-	-	-
EnSC	-	-	-	-	-
GSC	7.27	9.05	9.72	12.78	16.54
TSC	14.29	18.20	16.73	25.80	27.42
EKSS	3.27	5.83	5.62	14.96	8.26
Yale					
SSC-ADMM	-	-	-	-	-
SSC-OMP	14.80	14.80	79.81	79.81	79.81
EnSC	19.28	19.78	61.43	21.46	21.46
GSC	21.63	22.04	54.24	55.59	54.24
TSC	22.78	22.78	79.81	40.95	47.62
EKSS	16.04	25.16	78.50	78.50	78.50
COIL-20					
SSC-ADMM	13.19	15.28	16.32	60.49	13.19
SSC-OMP	28.40	28.40	31.18	66.18	28.40
EnSC	8.26	8.47	16.94	8.26	8.26
GSC	0.76	2.99	15.00	62.50	0.76
TSC	15.62	20.42	15.83	37.78	37.78
EKSS	5.28	7.43	7.43	21.53	21.53

Table 5.6: Performance of CQMs on common benchmark datasets known to have strong union of subspace structure. Results for SSC-OMP and EnSC not reported for Hopkins due to known poor performance on this dataset. Results for SSC-ADMM not reported on Yale due to high computational complexity of this algorithm; best known clustering error is 31.03%.

the noiseless case, the results indicate that perturbations such as those appearing in real datasets can confound this quality measure. On both the Hopkins and Yale datasets, graph-based CQMs fail to select good clusterings in all but a very few cases. The results on the COIL-20 database indicate the dependence of the graph-based methods (namely modularity) on the sparsity of the affinity matrix. While the KSS cost performs fairly well on this dataset, the best clusterings are selected by modularity for all variants of SSC and GSC. This is likely due to the fact that these methods result in especially sparse affinity matrices for this dataset, in contrast to TSC and EKSS, which do not necessarily perform best when the affinity matrix is sparse. Hence, when selecting CQMs, the user should be careful to consider the *type* of output typically provided by the algorithm being used. However, the continued strong performance of the KSS cost indicates that it is a reasonable selection for a general-purpose CQM for the problem of subspace clustering.

5.5 Axiomatic Study of Quality Measures

In this section, we discuss the axiomatic study of clustering quality. This topic differs significantly from the general study of clustering quality, and hence we provide a separate description of related work here. We define three axioms common to the study of CQMs in the context of distance-based clustering. We also briefly mention existing work in graph-based clustering quality. Finally, we present a first take on these axioms in the context of subspace clustering. However, we note that the work on this topic is still nascent and leads to axioms that are somewhat trivial. The development of a full axiomatic framework for subspace clustering is an interesting topic for our future research.

5.5.1 Related Work

Before discussing the relevant literature in detail, we note that there is a large overlap in the study of both CQMs and *clustering functions*. Given a pseudometric $\delta : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_{\geq 0}$ satisfying all properties of a true metric except the triangle inequality, a clustering function $f(\mathcal{X}, \delta)$ maps data into partitions known as clusters. In contrast, a (distance-based) CQM is a nonnegative function $m(\mathcal{C}, \mathcal{X}, \delta)$ that measures the quality of the resulting clustering. However, the formal study of what entails a good clustering function is essentially the same as that for a good CQM, and hence we discuss existing work on both topics. For the sake of understanding the general topic of "axiomatic clustering quality," the reader can view the two terms as interchangeable.

An early attempt to define algorithm-agnostic properties of a "good" clustering is given in [143]. The author proposes three axioms that any clustering function should satisfy and proves that, for the case where the number of clusters is not fixed, no function can simultaneously satisfy all three. These axioms were first considered in the context of CQMs in [25]; we now define these axioms and provide a bit of intuition behind each. Let m be a nonnegative quality measure over $(\mathcal{C}, \mathcal{X}, \delta)$, where we assume a higher value indicates a better clustering. The first axiom is known as *scale invariance* and states that if all distances between points are scaled *by the same constant*, the clustering quality should remain the same. Such a scaling could arise if the data are normalized by the maximum variance among all points.

Definition 5.1 (Scale Invariance [25])**.** *A quality measure m satisfies* scale invariance *if for every clustering $\mathcal{C}$ of $(\mathcal{X}, \delta)$, and every positive λ, $m(\mathcal{C}, \mathcal{X}, \delta) = m(\mathcal{C}, \mathcal{X}, \lambda\delta)$.*

A weaker definition of scale invariance was defined in [135] in the context of graph CQMs and requires only that the ordering of CQMs, rather than the exact value, be preserved under scaling.

Definition 5.2 (Relative Scale Invariance)**.** *A quality measure m satisfies* relative scale invariance *if for all clusterings $\mathcal{C}_1, \mathcal{C}_2$ of $(\mathcal{X}, \delta)$, and every positive λ, $m(\mathcal{C}_1, \mathcal{X}, \delta) \geq m(\mathcal{C}_2, \mathcal{X}, \delta)$ if and only if $m(\mathcal{C}_1, \mathcal{X}, \lambda\delta) \geq m(\mathcal{C}_2, \mathcal{X}, \lambda\delta)$.*

We will use this weaker notion of scale invariance in our UoS axioms in Section 5.5.2. The second axiom introduced by [25] is that of *consistency*, which loosely states that reducing intra-cluster distances and increasing inter-cluster distances does not decrease clustering quality. Intuitively, reducing the former type of distance should improve within-cluster similarity, while increasing the latter should improve separation between clusters. Before providing a formal definition, we first introduce the notion of a *consistent variant*.

Definition 5.3 (Consistent Variant [25])**.** *Given a clustering $\mathcal{C}$ over $(\mathcal{X}, \delta)$, a distance function δ' is a* $\mathcal{C}$-consistent variant *of δ if $\delta'(x, y) \leq \delta(x, y)$ for all $x \sim_c y$ and $\delta'(x, y) \geq \delta(x, y)$ for all $x \not\sim_c y$.*

Definition 5.4 (Consistency [25])**.** *A quality measure m satisfies* consistency *if for every clustering $\mathcal{C}$ over $(\mathcal{X}, \delta)$, whenever δ' is a $\mathcal{C}$-consistent variant of δ, then $m(\mathcal{C}, \mathcal{X}, \delta') \geq m(\mathcal{C}, \mathcal{X}, \delta)$.*

The final axiom, *richness*, rules out trivial CQMs by stating that for any clustering of the data, there exists a distance function that minimizes the quality measure.

Definition 5.5 (Richness [25])**.** *A quality measure m satisfies* richness *if for each nontrivial clustering $\mathcal{C}$ of $\mathcal{X}$, there exists a distance function δ such that $\mathcal{C} = \arg\max \{m(\mathcal{C}, \mathcal{X}, \delta)\}$.*

To see why richness is necessary, note that the trivial CQM $m(\mathcal{C}, \mathcal{X}, \delta) = 1$ satisfies both scale invariance and consistency. The authors of [25] show that these three axioms together form a consistent set by defining a number of CQMs that satisfy all three. However, we note that in this and other related studies [144, 145], the distance function provided to the CQM is *independent of the input clustering*. For example, to prove consistency, the authors of [25] choose a metric such that $\delta(x, y) = 1$ for all $x \sim_c y$ and $\delta(x, y) = 10$ otherwise. This assumption is reasonable for the general clustering case, where one may wish to compare clusterings obtained via different pseudometrics (or other means) under a distinct pseudometric that is independent of all others. Contrasting this, in the UoS case there is a clear notion of distance that is inherently tied to the given clustering, and hence these axioms must be modified if they are to be useful in defining valid CQMs for subspace clustering.

In [144], the authors also aim to resolve the impossibility theorem from [143] by relaxing the richness axiom to K-richness. The authors show that K-means satisfies scale invariance, consistency, and K-richness (note that this is in terms of *clustering functions*, and hence scale invariance is satisfied by the K-means cost, as linear scaling does not affect cluster assignments). The authors also introduce two new axioms that uniquely characterize the single linkage partitioning function [137, Ch. 14]. The first of these is *order consistency*, which states that for the graph whose edges are the distances between points, the resulting clustering is only a function of the *ordering* of the weights, rather than the weight values themselves. Distance-based clustering functions such as K-means do not satisfy order consistency. The final axiom included in [144] is that of *minimum*

spanning tree coherence (MST coherence), which states that two datasets with the same minimum spanning tree should return the same clustering. However, while the axioms presented provide insight into the general clustering problem, their main goal is to characterize single linkage, and hence the resulting axioms are not necessarily valuable for the purpose of empirically evaluating cluster quality across different algorithms.

A variety of other axioms are introduced throughout the literature on clustering quality. In [144], the authors introduce axioms that uniquely characterize the single linkage partitioning function [137, Ch. 14], with the goal of systematically distinguishing the properties of this clustering method from others and providing a means to organize different clustering paradigms. In [145], the authors revisit the seminal work of [143] and point out a number of implicit assumptions that lead to Kleinberg's impossibility result. Axiomatic CQMs for graph clustering are studied in [135], where the authors reformulate existing axioms and propose two new axioms specific to the graph clustering problem. The authors also show that modularity does not satisfy consistency (among other axioms), and develop a new CQM termed *adaptive scale modularity*, which is a generalization of modularity that does satisfy all proposed axioms. More recently, the authors of [136] propose a new graph-based CQM that accounts for costs associated with combining or splitting the input clusters. They analyze this CQM and show that it satisfies the axioms from [135].

5.5.2 Modified Axioms

While the axiomatic study of clustering quality continues to be a topic of interest within the clustering community, none of the proposed axioms or measures are applicable to data generated from a union of subspaces. We now turn our attention to a first attempt at resolving this problem by redefining the above three axioms in the context of subspace clustering. We consider only the case of noiseless data that are truly drawn from a union of subspaces. We also assume the number of subspaces K is known in advance. While these assumptions are quite strong compared to those in existing subspace clustering analyses, they allow us to define properties of a "good" CQM more clearly; the extension of the results from this section to the more general case is an interesting topic for future research. Finally, we note that the requirement $1 < K < N$ implies that the subspaces are *distinct* in the sense that $\overline{\text{aff}} < 1$.

As in Section 5.3, we base our axioms on distances from points to subspaces, where the subspaces are those resulting by performing PCA on the data from each cluster. We now redefine the axioms of relative scale invariance, consistency, and K-richness (richness with fixed K) under the UoS model. Recall that a CQM for UoS data is a function of the triplet $(\mathcal{C}, \mathcal{X}, \mathcal{U})$.

We begin with scale invariance, where we follow the definition of [135] and require the preservation of ordering under scaling, rather than exact equality. In the UoS setting, the distance considered

is a function of both the points themselves and the clusters (via the subspace bases, see Eq. (5.1)). Hence, it could be the case that the distance is scaled due to a reassignment of points to different clusters, which is a property that should not be captured by scale invariance. To overcome this issue, we redefine scale invariance such that only a scaling of the data themselves preserves CQM ordering.

Definition 5.6 (UoS Scale Invariance)**.** *A quality measure m satisfies* UoS scale invariance *if for all clusterings $\mathcal{C}_1, \mathcal{C}_2$ of $\mathcal{X}$, and every positive λ, $m(\mathcal{C}_1, \mathcal{X}, \mathcal{U}_1) \geq m(\mathcal{C}_2, \mathcal{X}, \mathcal{U}_2)$ if and only if $m(\mathcal{C}_1, \lambda\mathcal{X}, \mathcal{U}_1) \geq m(\mathcal{C}_2, \lambda\mathcal{X}, \mathcal{U}_2)$.*

We next turn our attention to consistency. Intuitively, a subspace CQM should improve if either (a) points are drawn closer to their assigned subspaces or (b) points are pushed farther from subspaces other than their closest. This idea is captured in UoS consistency, defined below.

Definition 5.7 (UoS Consistency)**.** *A quality measure m satisfies* UoS consistency *if the following are true for any $x \in c_k$.*

- *The quality measure m is a nonincreasing function of $\operatorname{dist}(x, \mathcal{S}_k)$.*
- *The quality measure m is a nondecreasing function of $\operatorname{dist}(x, S_j)$ for all $j \neq k$.*

One key difference between the notion of consistency introduced here and that from [25] is that the above allows for changes in individual distances from points to subspaces, rather than requiring that all distances be changed. This is a necessary property in the UoS case, as improvements in subspace estimates may reduce the distances for some points in a subspace, while leaving others unchanged. For example, two points may lie along orthogonal directions within a subspace, and hence a change in the corresponding basis along one of these directions would only affect the distance for the corresponding point.

Finally, we define the analog of richness for the UoS case. Since we assume the number of clusters is known, we borrow the notation from [144] and refer to this axiom as UoS K-richness. This axiom differs from existing notions of richness significantly since we fix our metric to be that of Eq. (5.1). The key difference stems from the fact that the distance we consider is that from a point to a subspace, and hence it is not reasonable to expect *any* possible clustering to be captured by this distance, which corresponds to a strong geometric model. Instead, we require that our CQM should be flexible enough to capture *any subspace arrangement* and strict enough to be optimal for the *correct clustering of points according to their true subspaces*. As we will show shortly, this notion of richness still rules out trivial CQMs but also rejects possible CQMs derived from existing subspace clustering algorithms.

Definition 5.8 (UoS K-Richness)**.** *Let $\mathcal{X}$ be a collection of points belonging to K subspaces spanned by $\mathcal{U}^* = \{U_1, \cdots, U_K\}$. A quality measure m satisfies* UoS K-richness *if for each arrangement of subspaces $\mathcal{U}^*$, the true clustering $\mathcal{C}^*$ is such that $\mathcal{C}^* = \arg\max \{m(\mathcal{C}^*, \mathcal{X}, \mathcal{U}^*)\}$.*

First note that trivial functions such as the constant CQM are ruled out by K-richness. Next, we note the importance of requiring optimality for the correct clustering. Given that we propose the use of the KSS cost function as a CQM above, it is reasonable to ask whether other cost functions could also be applied. For example, the TSC hinges on the idea that intra-cluster inner products should exceed inter-cluster inner products, so a natural CQM based on this principle could be

$$m_{\text{TSC}}(\mathcal{C}, \mathcal{X}, \mathcal{U}) = \sum_{k=1}^{K} \sum_{x_i, x_j \in c_k} \left| x_i^T x_j \right|.$$

However, if the subspaces are sufficiently close in terms of their principal angles or points happen to be concentrated near the intersection of subspaces, all inner products will be roughly the same. Hence, we cannot guarantee that the above will be optimal for *every* correct clustering of the data (where a "correct" clustering is one in which points are clustered with all others generated from the same subspace). In contrast, the KSS cost (5.2) is always zero given the correct clustering of the data.

5.5.3 Analysis of Proposed Axioms

We now analyze our proposed CQMs show that KSS cost and subspace margin are consistent with all proposed axioms above, and normalized margin is scale invariant and K-rich. The question of whether normalized margin is consistent is an open topic for our continued study. Since all three proposed CQMs are similarly formulated, we verify each axiom for all CQMs simultaneously. Finally, we discuss a shortcoming of the given axioms and discuss ideas for an additional axiom to be developed in future work.

Proposition 5.1. *KSS cost, subspace margin, and normalized margin are UoS scale invariant.*

Proof. Note that $\operatorname{dist}(\lambda x, \mathcal{S}) = \lambda \operatorname{dist}(x, \mathcal{S})$. Hence,

$$m_{\text{KSS}}(\mathcal{C}, \lambda\mathcal{X}, \mathcal{U}) = \lambda^2 m_{\text{KSS}}(\mathcal{C}, \mathcal{X}, \mathcal{U}),$$

and

$$m_{\text{KSS}}(\mathcal{C}_1, \mathcal{X}, \mathcal{U}) \geq m_{\text{KSS}}(\mathcal{C}_2, \mathcal{X}, \mathcal{U}) \Leftrightarrow \lambda^2 m_{\text{KSS}}(\mathcal{C}_1, \mathcal{X}, \mathcal{U}) \geq \lambda^2 m_{\text{KSS}}(\mathcal{C}_2, \mathcal{X}, \mathcal{U}).$$

Therefore, KSS cost is scale invariant. By the same principle,

$$m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \lambda\mathcal{U}) = m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \mathcal{U})$$

and subspace margin is scale invariant. Finally, note that the estimated subspace basis for cluster k is

$$U_k = \underset{U \in \text{St}(D,d)}{\arg\max} \left\| U^T X_k \right\|_F^2, \tag{5.5}$$

where $\text{St}(D,d) = \left\{ U \in \mathbb{R}^{D \times d} : U^T U = I_d \right\}$ and X_k is the matrix whose columns are the points in c_k. From (5.5), we see that scaling all points by the same constant does not affect the subspace estimates. Hence, the quantity $\overline{\text{aff}}$ does not depend on the scaling, so normalized margin is also scale invariant. □

Proposition 5.2. *KSS cost and subspace margin are UoS consistent.*

Proof. Let $\mathcal{C}$ be a clustering over $\mathcal{X}$. For any $x \in c_k$, decreasing $\text{dist}(x, \mathcal{S}_k)$ decreases $m_{\text{KSS}}(\mathcal{C}, \mathcal{X}, \mathcal{U})$. Similarly, for any $x_i \in c_{k(i)}$, decreasing $\text{dist}(x(i), \mathcal{S}_{k(i)})$ decreases the numerator of $m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \mathcal{U})$. Likewise, increasing $\text{dist}(x(i), \mathcal{S}_{k'(i)})$ increases the denominator of $m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \mathcal{U})$. Both changes result in a reduction in $m_{\text{margin}}(\mathcal{C}, \mathcal{X}, \mathcal{U})$. □

As mentioned, it is not clear whether normalized margin is UoS consistent, since changes in $\text{dist}(x, \mathcal{S})$ also change the subspace estimates. In cases where the true subspaces are close, this could increase $\overline{\text{aff}}$, consequently increasing normalized margin. Determining whether this change is balanced by the reduced value of $\text{dist}(x, \mathcal{S})$ is an interesting question for our future research.

Proposition 5.3. *KSS cost, subspace margin, and normalized margin are UoS K-rich.*

Proof. Note that by assumption $\overline{\text{aff}} < 1$. Given any arrangement of data across K subspaces, all three CQMs attain a value of zero under the correct clustering. □

We now note a shortcoming with the proposed axioms, namely that none of them captures the dependence on the true subspace dimension. In particular, while UoS richness requires that the optimal value be attained for the true clustering and subspace bases, it does not state that the CQM be optimal *only* in the case of the true clustering and subspace bases. Consider the following data arrangement. Let $N_k > 2$ points be drawn uniformly at random from each of $K = 2$ subspaces of dimension $d = 2$ in ambient dimension $D = 1000$, and set $\text{aff}(\mathcal{S}_1, \mathcal{S}_2) = 0$ (orthogonal subspaces). In this case, the true subspace bases result in a value of zero for all three proposed CQMs. However, this cost is also attainable for estimated subspaces of dimension as high as 500 by appending vectors from the null space of the two subspaces, even for the normalized margin CQM. This indicates the need for at least one additional axiom that does not allow arbitrarily high subspace dimensions,

e.g., requiring that the sum of subspace dimensions be no more than the rank of the data matrix. In our future work on this topic, we plan to incorporate an axiom to capture this important feature of UoS data.

5.6 Discussion on Subspace Models

The above study of both empirical and axiomatic clustering quality hinges on the idea that the underlying subspaces cluster the data in a meaningful way. However, it may be the case that even after developing a notion of a "best" CQM for subspace clustering, a high-quality clustering of data according to the UoS model does not correspond to information that is useful to practitioners. Note that this problem is not unique to the UoS model. For example, clustering the Yale face database according to Euclidean distance results in clusters that correspond to lighting condition, i.e., all faces with large amount of shadow are clustered together. In contrast, the UoS model clusters the images according to subject, providing information that is useful for the desired task. The problem of automatically determining which model is appropriate for a given goal is a difficult topic that requires further study. Progress on this topic will rely heavily on strong interplay between practitioners and developers of clustering algorithms. For a further discussion related to this topic, see [132].

5.7 Conclusion

As subspace clustering algorithms rely heavily on parameter selection, the appropriate choice of CQM is tantamount to the utility of subspace clustering methods in real-world situations. To the best of our knowledge, no principled study of parameter selection has been performed to this point. In this chapter, we proposed and examined three measures of clustering quality specific to the problem of subspace clustering. The proposed methods outperform existing CQMs in terms of selecting the parameters that achieve the lowest clustering error for subspace clustering algorithms. In the case where the subspace dimensions are unknown, the normalized margin measure is capable of selecting the appropriate subspace dimension. We also briefly discussed and extended an axiomatic framework for studying clustering quality and analyzed our axioms from this perspective.

As this is ongoing work, a number of open questions remain. First, while the proposed CQMs outperform existing measures in the majority of simulations, they often fail to capture the "optimal" parameters, in the sense that they do not select the parameters corresponding to the lowest clustering error. Therefore, the continued development of CQMs for subspace clustering data is an important open problem. One potential avenue toward this end would be to extend existing CQMs by replacing pairwise distances with pairwise absolute inner products. It is well-known [41] that points within

a subspace have larger pairwise inner products than points in distinct subspaces, as long as the subspaces are not too close in terms of their affinity. Another open question is whether there are any axioms in addition to those stated in Section 5.5.2 that would better characterize CQMs for the subspace clustering problem. A number of graph-specific axioms have been proposed [135], and hence it is natural to ask whether any of these can provide insights into new axioms for subspace clustering. Finally, the experiments on real data indicate some weaknesses in the proposed methods in the case where the data are not truly generated from a union of subspaces. One potential avenue for future research is that of developing robust CQMs that allow for small nonlinearities in the geometric structures, i.e., considering the data as drawn from a union of manifolds. Another avenue stems from the fact that while we have made strong use of the underlying geometric structure in the data, the proposed CQMs ignore the resulting graph structure provided by the various algorithms. It is likely that a CQM which jointly considers both the geometric and the graph structure would provide superior performance across all datasets.

CHAPTER 6

Conclusion & Future Work

In this thesis, we have presented a number of ways to overcome the challenges of modern data analysis problems by leveraging key features of high-volume and high-dimensional data. We have presented novel algorithms and analysis for two important problems in signal processing, using tools from machine learning, high-dimensional probability, and linear algebra. We now conclude with a brief summary of each chapter and a discussion of the future work proposed within each chapter.

6.1 Active Learning for Spatial Sampling

In Chapter 2, we developed and analyzed a novel active learning algorithm for determining the threshold of one-dimensional step functions that balances the number of samples taken and distance traveled throughout the estimation procedure. We showed how these one-dimensional estimators can be combined to efficiently determine the spatial extent of a hypoxic region in lakes of interest and demonstrated the advantages of our method through simulations and real-world experiments.

Several open questions are discussed in Section 2.5 of the chapter. Perhaps the most impactful of these would be the extension of our method to arbitrary boundary classes, whether via the mentioned S^2 algorithm [70] or some other means. Another important question is that of optimality for penalized algorithms such as quantile search. Existing tools for such an analysis include dynamic programming [146, Ch. 6] and adaptive submodularity [147]. However, these approaches only apply to *greedy* algorithms, such as proactive learning. Developing a means to provide useful analysis for *any* active learning algorithm with non-uniform costs would be of great practical and theoretical interest.

6.2 Active Learning for Subspace Clustering

In Chapter 3, we presented a means of incorporating ideas from active learning into the problem of subspace clustering by performing actively selected pairwise comparisons between points. We defined a notion of margin specific to the union-of-subspaces model and showed that points lying near the intersection of subspaces will be those of minimum subspace margin with high probability. Finally, we showed through simulations on a variety of benchmark datasets that our method dramatically outperforms existing algorithms on data from a union of subspaces and does not seem to perform any worse on data with no known UoS structure.

A number of exciting directions for future work are discussed within the chapter. Particularly intriguing is the extension of SUPERPAC to arbitrary geometric structures. One may notice that the certain sets formed by our algorithm are equivalently sets of labeled data; hence, it may be reasonable to simply train a classifier of choice (e.g., SVM or random forest) on this labeled data and use it to calculate margin. This leads to the broader question of the difference between pairwise constrained clustering and semi-supervised learning. Given enough pairwise comparisons, it may be advantageous to switch from obtaining labels via clustering techniques to some semi-supervised classifier. Discovering the appropriate boundary between these methods is an interesting open question that has received no attention to the best of our knowledge.

6.3 Ensemble Methods for Subspace Clustering

In Chapter 4, we demonstrated that many inaccurate but computationally-efficient base clusterings can be intelligently combined to achieve state-of-the-art performance in subspace clustering. We analyzed a simplified version of the K-subspaces algorithm and showed that our proposed ensemble approach exhibits the same theoretical guarantees as existing subspace clustering algorithms. We also presented several heuristics for improving the clustering performance of our algorithm and demonstrated empirical success across a wide variety of benchmark datasets.

While there are many important directions for future work on this topic, the most broadly applicable is that of developing a general framework for ensembles of subspace clustering algorithms. Existing consensus clustering literature such as [16, 118] provides strong empirical study of these methods but falls short of developing guarantees for the output clustering. We showed in Chapter 4 that the model assumptions of subspace clustering unlock these guarantees in the specific case where KSS is used as the base classifier. It would be interesting to characterize which, if any, other algorithms will exhibit improved performance when used in the consensus clustering framework.

6.4 Clustering Quality Measures for Subspace Clustering

In Chapter 5, we studied the application of clustering quality measures to the subspace clustering problem. As existing subspace clustering algorithms rely heavily on the correct choice of parameters, the ability to select these parameters in the absence of ground-truth labels is paramount to the applicability of this model. We briefly described why existing CQMs perform poorly on UoS data and proposed novel CQMs that are capable of selecting both the underlying subspace dimension and appropriate algorithm parameters. We also presented a first take on an axiomatic analysis of this problem.

A number of open questions will be addressed as this ongoing work continues. Among these, the most intriguing are combining graph-based and UoS-based CQMs and the further development of axioms for subspace clustering quality. As the dependence on spectral clustering is a continuing trend in subspace clustering, it is important to understand the connection between existing graph-based CQMs and the various subspace clustering algorithms one may employ. Further, graph-based methods do not rely on geometric assumptions, making them a better choice in the case where the data do not fit the UoS assumption. Finally, it would be of theoretical and practical interest to create a working set of axioms for CQMs specific to the UoS data model. These axioms would serve primarily to aid the design of new CQMs for subspace clustering but would also have the potential to inform the development of new subspace clustering algorithms by giving a clearer picture of the attributes of a "good" algorithm.

APPENDIX A

Proofs for Quantile Search

Included in this appendix are the proofs of Thms. 2.1-2.3 of Chapter 2. We also include a proof that the truncated probabilistic quantile search algorithm satisfies the statement of Thm. 2.3.

A.1 Deterministic Quantile Search

In this section, we provide the full proofs of Thms. 2.1 and 2.2 of Chapter 2. We provide the DQS algorithm in Algorithm A.1 for reference.

Theorem A.1. *Consider a deterministic quantile search with parameter m and let $\rho = \frac{m-1}{m}$. Begin with a uniform prior on θ. The expected estimation error after n measurements is then*

$$\mathbb{E}[|\hat{\theta}_n - \theta|] = \frac{1}{4}\left[\rho^2 + (1-\rho)^2\right]^n. \tag{A.1}$$

Proof. The proof follows by induction. After n samples, the unit interval has been split into 2^n subintervals, one for each possible sequence of n measurements. In order to find the expected error, we break the interval down by subinterval, find the conditional expected error given that θ is in each subinterval, and combine all subintervals using the law of total expectation. We note that in binary search, these subintervals are all the same length, but in quantile search each has a different length.

Let Z_n denote the error after n samples, e.g., $Z_n = |\hat{\theta}_n - \theta|$. We establish the base case by direct computation. The first sample is made at $1/m = 1 - \rho$ into the interval. The expected error after one sample is then

$$\begin{aligned}\mathbb{E}[Z_1] &= \mathbb{E}\left[Z_1 | \theta \leq \frac{1}{m}\right]\mathbb{P}\left(\theta \leq \frac{1}{m}\right) + \mathbb{E}\left[Z_1 | \theta > \frac{1}{m}\right]\mathbb{P}\left(\theta > \frac{1}{m}\right) \\ &= \frac{1}{4}\left[(1-\rho)^2 + \rho^2\right].\end{aligned}$$

After the second measurement, each interval $[0, 1-\rho]$ and $[1-\rho, 1]$ is split into two subintervals,

Algorithm A.1 Deterministic Quantile Search (DQS)

Input: search parameter m, sample budget N
Initialize: $X_0 \leftarrow 0$, $Y_0 \leftarrow 1$, $n \leftarrow 1$, $a \leftarrow 0$, $b \leftarrow 1$
while $n \leq N$ **do**
  if $Y_{n-1} = 1$ **then**
    $X_n \leftarrow X_{n-1} + \frac{1}{m}(b-a)$
  else
    $X_n \leftarrow X_{n-1} - \frac{1}{m}(b-a)$
  end if
  $Y_n \leftarrow f(X_n)$
  $a = \max\{X_i : Y_i = 1, i \leq n\}$
  $b = \min\{X_i : Y_i = 0, i \leq n\}$
  $\hat{\theta}_n \leftarrow \frac{a+b}{2}$
end while

and the error can be calculated again using the law of total expectation. We generalize this idea using the following lemma.

Lemma A.1. *Suppose at sample n we know θ lies in a subinterval $[a,b] \subset [0,1]$ with normalized conditional error $\mathbb{E}[Z_n|\theta \in [a,b]]\mathbb{P}(\theta \in [a,b]) = \frac{1}{4}\rho^{2i}(1-\rho)^{2j}$. At sample $n+1$, we split $[a,b]$ into two subintervals according to quantile search. Then the normalized conditional error of one subinterval will be $\frac{1}{4}\rho^{2i}(1-\rho)^{2j+2}$ and of the other will be $\frac{1}{4}\rho^{2i+2}(1-\rho)^{2j}$.*

Proof. Note that quantile search either splits the interval at the point $a + \frac{1}{m}(b-a)$ or $b - \frac{1}{m}(b-a)$, depending on the value of Y_n. We show the first case (the second is symmetric). At sample $n+1$, we have

$$\mathbb{E}[Z_{n+1}|\theta \in [a,b]] = \frac{1}{4}(b-a)^2(1-\rho)^2 + \frac{1}{4}(b-a)^2\rho^2.$$

We now show that $(b-a)^2 = \rho^{2i}(1-\rho)^{2j}$ using induction on n. Consider the base case of $n=1$ and note that $1/m = 1-\rho$. Then the possible intervals are $[0, 1/m]$ and $[1/m, 1]$, and we have

$$(b-a)^2 = \begin{cases} (1-\rho)^2, & [a,b] = [0, \frac{1}{m}] \\ \rho^2, & [a,b] = [\frac{1}{m}, 1] \end{cases}.$$

Noting that $\mathbb{E}[Z_1] = \frac{1}{4}[(1-\rho)^2 + \rho^2]$ proves the base case. Now suppose that $(b-a)^2 = \rho^{2i}(1-\rho)^{2j}$ for some n such that $\mathbb{E}[Z_n|\theta \in [a,b]]\mathbb{P}(\theta \in [a,b]) = \frac{1}{4}\rho^{2i}(1-\rho)^{2j}$. Splitting the interval at $a + \frac{1}{m}(b-a)$ yields two subintervals, $[c,d]$ and $[e,f]$ with

$$\begin{aligned} (d-c)^2 &= \left(a + \frac{1}{m}(b-a) - a\right)^2 \\ &= (1-\rho)^2\rho^{2i}(1-\rho)^{2j} \end{aligned}$$

and

$$\begin{aligned}(f-e)^2 &= \left(b-a-\frac{1}{m}(b-a)\right)^2 \\ &= \rho^2\rho^{2i}(1-\rho)^{2j}.\end{aligned}$$

Therefore we see that

$$\mathbb{E}[Z_{n+1}|\theta\in[a,b]] = \frac{1}{4}\rho^{2i}(1-\rho)^{2j+2} + \frac{1}{4}\rho^{2i+2}(1-\rho)^{2j}$$

and the proof is complete. □

We now use this lemma to prove the induction step. Suppose at sample n we have

$$\mathbb{E}[Z_n] = \frac{1}{4}\sum_{k=0}^{n}\binom{n}{k}\rho^{2(n-k)}(1-\rho)^{2k}\ .$$

With the next sample, the intervals will split into two. We then apply the lemma to see that the resulting conditional expectation will be

$$\begin{aligned}\mathbb{E}[Z_{n+1}] &= \frac{1}{4}\sum_{k=0}^{n}\binom{n}{k}\left(\rho^{2(n-k)+2}(1-\rho)^{2k} + \rho^{2(n-k)}(1-\rho)^{2k+2}\right) \\ &= \frac{1}{4}\binom{n+1}{0}\rho^{2(n+1)} + \frac{1}{4}\sum_{k=1}^{n}\left(\binom{n}{k}+\binom{n}{k-1}\right)\rho^{2(n+1-k)}(1-\rho)^{2k} + \\ &\quad \frac{1}{4}\binom{n}{n}(1-\rho)^{2(n+1)}.\end{aligned}$$

Using the fact that $\binom{n}{n} = \binom{n+1}{n+1} = 1$ and

$$\binom{n}{k}+\binom{n}{k-1} = \binom{n+1}{k},$$

we have

$$\mathbb{E}[Z_{n+1}] = \frac{1}{4}\sum_{k=0}^{n+1}\binom{n+1}{k}\rho^{2(n+1-k)}(1-\rho)^{2k}\ .$$

Recalling the binomial formula

$$(x+y)^n = \sum_{k=0}^{n}\binom{n}{k}x^{n-k}y^k$$

completes the proof.

□

Theorem A.2. *Let D denote a random variable representing the distance traveled during a deterministic quantile search with parameter m. Begin with a uniform prior on θ. Then*

$$\mathbb{E}[D] = \frac{m}{2m-2}. \tag{A.2}$$

Proof. Following the proof sketch given in Chapter 2, we first rewrite the interval A_i by simplifying the geometric sums. Let $\rho = (m-1)/m$. Then

$$\begin{aligned} A_i &= \left[\frac{1}{m}\frac{\rho^i - 1}{\rho - 1}, \frac{1}{m}\frac{\rho^{i+1} - 1}{\rho - 1}\right) \\ &= \left[1-\rho^i, 1-\rho^{i+1}\right), \end{aligned}$$

where we have used the fact that $1/m = 1 - \rho$. It is tedious but straightforward to define the subintervals that partition A_i, and the following result is obtained.

$$A_{i_1 i_2} = \left((1-\rho^{i_1+1}) - \frac{1}{m}\rho^{i_1}(1-\rho^{i_2+1}), (1-\rho^{i_1+1}) - \frac{1}{m}\rho^{i_1}(1-\rho^{i_2})\right]$$

After further inspection, one can obtain a general equation for odd values of n (and a similar equation for even values). The DQS algorithm for our class of functions chooses samples moving further into the interval until it makes a zero measurement. It then turns back and takes samples in the opposite direction until a one is measured. This behavior allows us to analyze the total distance by splitting it up into stages—first the expected distance traveled before the algorithm reaches a point $x_1 > \theta$, and then $x_2 < \theta$, etc. Let D_n be a random variable denoting the distance required to move to the right of θ for the $\left\lceil \frac{n}{2} \right\rceil$th time when n is odd, and to the left of θ for the $\frac{n}{2}$th time when n is even. Then by linearity of expectation, the total expected distance is

$$\mathbb{E}[D] = \sum_{n=1}^{\infty} \mathbb{E}[D_n]. \tag{A.3}$$

We now calculate $\mathbb{E}[D_n]$. Using the above definition, we can easily calculate $\mathbb{E}[D_n | \theta \in A_{i_1 \ldots i_n}]$ by taking the absolute value of the difference between the lower edge of $A_{i_1 \ldots i_{n-1}}$ and the upper edge of $A_{i_1 \ldots i_n}$ for odd values of n (and the converse for even values). This yields

$$\mathbb{E}[D_n | \theta \in A_{i_1 \ldots i_n}] = \frac{1}{m^{n-1}}\rho^{i_1 + \cdots + i_{n-1}}\left(1 - \rho^{i_n+1}\right).$$

The probability $\mathbb{P}\left(\theta \in A_{i_1 \ldots i_n}\right)$ is easily calculated as the length of the interval. Note that the

majority of terms cancel, so the result becomes

$$\begin{aligned}\mathbb{P}\left(\theta \in A_{i_1 \ldots i_n}\right) &= \frac{1}{m^{n-1}} \rho^{i_1+\cdots+i_{n-1}} \left(\rho^{i_n} - \rho^{i_n+1}\right) \\ &= \frac{1}{m^n} \rho^{i_1+\cdots+i_n}.\end{aligned}$$

We now prove the expected distance using induction on $\mathbb{E}[D_n]$. The base case has been shown to be

$$\mathbb{E}[D_1] = \frac{m}{2m-1}.$$

Now assume for arbitrary n that

$$\mathbb{E}[D_{n-1}] = \frac{m}{(2m-1)^{n-1}}.$$

Using the above definitions, we see that

$$\begin{aligned}\mathbb{E}[D_n] &= \sum_{i_1=0}^{\infty} \cdots \sum_{i_n=0}^{\infty} \mathbb{E}[D_n | \theta \in A_{i_1 \ldots i_n}] \mathbb{P}\left(\theta \in A_{i_1 \ldots i_n}\right) \\ &= \frac{1}{2m-1} \mathbb{E}[D_{n-1}] = \frac{m}{(2m-1)^n}.\end{aligned}$$

This shows the inductive hypothesis and may obtain $\mathbb{E}[D]$ by (A.3).

□

A.2 Probabilistic Quantile Search

In this section, we provide the full proof for Thm. 2.3. As stated in Chapter 2, we analyze a discretized version of PQS given in Algorithm A.2.

Theorem A.3. *Under the assumptions given in the case of noisy measurements, the discretized PQS algorithm satisfies*

$$\begin{aligned}\sup_{\theta \in [0,1]} \Pr\left(|\hat{\theta}_n - \theta| > \Delta\right) \leq \frac{1-\Delta}{\Delta} \Bigg[&\frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} \\ &+ \left(\frac{1-p}{2(1-\alpha)} - \frac{p}{2\alpha}\right)(1-2\alpha)\left(\frac{m-2}{m}\right)\Bigg]^n.\end{aligned}$$

Algorithm A.2 Discretized Probabilistic Quantile Search

Input: search parameter m, sample budget N, $\Delta > 0$ such that $\Delta^{-1} \in \mathbb{N}$, α, $\beta = 1 - \alpha$

Define: posterior after measurement j as $\pi_j : [0, 1] \to \mathbb{R}$

$$\pi_j(x) = \sum_{i=1}^{\Delta^{-1}} a_i(j) \mathbf{1}_{I_i}(x),$$

where $I_1 = [0, \Delta]$ and $I_i = (\Delta(i-1), \Delta i]$, for $i \in \{2, \ldots, \Delta^{-1}\}$ form a partition of the unit interval

Initialize: $a_i(0) = \Delta$ such that $\sum_{i=1}^{\Delta^{-1}} a_i(j) = 1$

while $n \leq N$ **do**

define $k(j) \in \left\{1, \ldots, \Delta^{-1}\right\}$ such that

$$\sum_{i=1}^{k(j)-1} a_i(j) \leq \frac{1}{m}, \qquad \sum_{i=1}^{k(j)} a_i(j) > \frac{1}{m}$$

compute $\tau_1(j)$, $\tau_2(j)$, $P_1(j)$, and $P_2(j)$ as

$$\begin{aligned}
\tau_1(j) &= \left[\sum_{i=k(j)}^{\Delta^{-1}} a_i(j) - \frac{m-1}{m}\right] - \left[\sum_{i=1}^{k(j)-1} a_i(j) - \frac{1}{m}\right] \\
\tau_2(j) &= \left[\sum_{i=1}^{k(j)} a_i(j) - \frac{1}{m}\right] - \left[\sum_{i=k(j)+1}^{\Delta^{-1}} a_i(j) - \frac{m-1}{m}\right] \\
P_1(j) &= \frac{\tau_2(j)}{\tau_1(j) + \tau_2(j)}
\end{aligned}$$

choose

$$X_{j+1} = \begin{cases} \Delta(k(j)-1), & \text{with probability } P_1(j) \\ \Delta k(j), & \text{with probability } P_2(j) = 1 - P_1(j) \end{cases}$$

and define $k = X_{j+1}\Delta^{-1}$ and

$$\tau = \sum_{i=1}^{k} a_i(j) - \sum_{i=k+1}^{\Delta^{-1}} a_i(j)$$

observe $Y_{j+1} = f(X_{j+1}) \oplus U_{j+1}$ where $U_{j+1} \sim \text{Bern}(\alpha)$

if $Y_{j+1} = 0$ **then**

$$a_i(j+1) = \begin{cases} \frac{2\beta}{1+(\tau-(m-2)/m)(\beta-\alpha)} a_i(j) & i \leq k \\ \frac{2\alpha}{1+(\tau-(m-2)/m)(\beta-\alpha)} a_i(j) & i > k \end{cases}$$

else

$$a_i(j+1) = \begin{cases} \frac{2\alpha}{1+((m-2)/m-\tau)(\beta-\alpha)} a_i(j) & i \leq k \\ \frac{2\beta}{1+((m-2)/m-\tau)(\beta-\alpha)} a_i(j) & i > k \end{cases}$$

end if

end while

estimate $\hat{\theta}_n$ such that $\sum_0^{\hat{\theta}_n} \pi_n(x) \approx 1/2$

Taking

$$\Delta^{-1} = \left[\frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} + \left(\frac{1-p}{2(1-\alpha)} - \frac{p}{2\alpha}\right)(1-2\alpha)\left(\frac{m-2}{m}\right)\right]^{-n/2}$$

yields a bound on the expected error

$$\sup_{\theta\in[0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq 2\left[\frac{1-p}{2(1-\alpha)} + \frac{p}{2\alpha} + \left(\frac{1-p}{2(1-\alpha)} - \frac{p}{2\alpha}\right)(1-2\alpha)\left(\frac{m-2}{m}\right)\right]^{n/2}.$$

Finally, taking $\alpha = \sqrt{p}/(\sqrt{p}+\sqrt{q})$ *minimizes the right hand side of these bounds, yielding*

$$\sup_{\theta\in[0,1]} \mathbb{E}[|\hat{\theta}_n - \theta|] \leq 2\left(\frac{m-1}{m} + \frac{2\sqrt{p(1-p)}}{m}\right)^{n/2}. \tag{A.4}$$

Proof. The proof follows that of [49, 6]. Recall from the discretized algorithm (Algorithm A.2 that I_i denotes the ith discretized bin of the interval and $a_i(j)$ denotes the posterior mass at the ith bin after j measurements. First define $k(\theta)$ to be the index of the bin I_i containing θ, so that $\theta \in I_{k(\theta)}$. Next define the following terms

$$M_\theta(j) = \frac{1 - a_{k(\theta)}(j)}{a_{k(\theta)}(j)},$$

and

$$N_\theta(j+1) = \frac{M_\theta(j+1)}{M_\theta(j)} = \frac{a_{k(\theta)}(j)(1 - a_{k(\theta)}(j+1)}{a_{k(\theta)}(j+1)(1 - a_{k(\theta)}j)}. \tag{A.5}$$

As a brief bit of intuition, note that $M_\theta(j)$ is a measure of how much mass is in the bin actually containing θ after the jth measurement, while $N_\theta(j+1)$ is a measure of improvement from one iteration to the next and is strictly less than one when an improvement (correct measurement) is made [6]. From [6] and following a straightforward application of Markov's inequality and the law of total expectation, we have that

$$\mathbb{P}\left(|\widehat{\theta}_n - \theta| > \Delta\right) \leq \mathbb{E}[M_\theta(n)], \tag{A.6}$$

and

$$\mathbb{E}[M_\theta(n)] \leq M_\theta(0)\left\{\max_{j\in 0,\ldots,n-1} \max_{\mathbf{a}(j)} \mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)]\right\}^n. \tag{A.7}$$

The remainder of the proof is to bound $\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)]$. To do this, we consider three cases: (i) $k(j) = k(\theta)$; (ii) $k(j) > k(\theta)$; and (iii) $k(j) < k(\theta)$. First, consider the case (i), where $k(j) = k(\theta)$,

and let $X_{j+1} = \Delta(k(j) - 1)$ so that the correct measurement is $Y_{j+1} = 1$. In this case, k in the algorithm is $k(\theta) - 1$, and hence $i = k + 1$ for updating. Also, assume the case where we measure correctly, so that with probability $q = 1 - p$

$$a_{k(\theta)}(j+1) = \frac{2\beta}{1 + ((m-2)/m - \tau)(\beta - \alpha)},$$

where $\beta = 1 - \alpha$. Then we have

$$\begin{aligned}
N_\theta(j+1) &= \frac{a_{k(\theta)}(j)\left(1 - \frac{2\beta}{1+((m-2)/m-\tau)(\beta-\alpha)} a_{k(\theta)}(j)\right)}{\frac{2\beta}{1+((m-2)/m-\tau)(\beta-\alpha)} a_{k(\theta)}(j)\left(1 - a_{k(\theta)}(j)\right)} \\
&= \frac{1 + ((m-2)/m - \tau)(\beta - \alpha) - 2\beta a_{k(\theta)}(j) + a_{k(\theta)}(j) - a_{k(\theta)}(j)}{2\beta\left(1 - a_{k(\theta)}(j)\right)} \\
&= \frac{1 - a_{k(\theta)}(j) + (\beta - \alpha)\left((m-2)/m - \tau - a_{k(\theta)}(j)\right)}{2\beta\left(1 - a_{k(\theta)}(j)\right)} \\
&= \frac{1 - a_{k(\theta)}(j)}{2\beta\left(1 - a_{k(\theta)}(j)\right)} + \frac{(\beta - \alpha)\left((m-2)/m - \tau - a_{k(\theta)}(j)\right)}{2\beta\left(1 - a_{k(\theta)}(j)\right)} \\
&= \frac{1}{2\beta} + \frac{(\beta - \alpha)x}{2\beta},
\end{aligned}$$

where

$$x = \frac{\tau_1(j) + (m-2)/m - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}$$

and $\tau_1(j)$ is defined in Algorithm A.2. Similarly, when $X_{j+1} = \Delta k(j)$,

$$\begin{aligned}
N_\theta(j+1) &= \frac{a_{k(\theta)}(j)\left(1 - \frac{2\beta}{1+(\tau-(m-2)/m)(\beta-\alpha)}a_{k(\theta)}(j)\right)}{\frac{2\beta}{1+(\tau-(m-2)/m)(\beta-\alpha)}a_{k(\theta)}(j)\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1 - \frac{2\beta}{1+(\tau-(m-2)/m)(\beta-\alpha)}a_{k(\theta)}(j)}{\frac{2\beta}{1+(\tau-(m-2)/m)(\beta-\alpha)}\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1+(\tau-(m-2)/m)(\beta-\alpha) - 2\beta a_{k(\theta)}(j)}{2\beta\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1+(\tau-(m-2)/m)(\beta-\alpha) - 2\beta a_{k(\theta)}(j) + a_{k(\theta)}(j) - a_{k(\theta)}(j)}{2\beta\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1+(\tau-(m-2)/m)(\beta-\alpha) - a_{k(\theta)}(j)(\beta-\alpha) - a_{k(\theta)}(j)}{2\beta\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1-a_{k(\theta)}(j) + (\beta-\alpha)\left(\tau-(m-2)/m - a_{k(\theta)}(j)\right)}{2\beta\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1-a_{k(\theta)}(j)}{2\beta\left(1-a_{k(\theta)}(j)\right)} + \frac{(\beta-\alpha)\left(\tau-(m-2)/m - a_{k(\theta)}(j)\right)}{2\beta\left(1-a_{k(\theta)}(j)\right)} \\
&= \frac{1}{2\beta} + \frac{(\beta-\alpha)x}{2\beta},
\end{aligned}$$

where

$$x = \frac{\tau_2(j) - (m-2)/m - a_{k(\theta)}(j)}{1 - a_{k(\theta)}(j)}$$

and $\tau_2(j)$ is defined in Algorithm A.2. Equivalent steps can be followed for cases (ii) and (iii), yielding (respectively)

$$x = \begin{cases} \frac{-\tau_1(j)-(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta(k(j)-1) \\ \frac{\tau_2(j)-(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta k(j) \end{cases},$$

and

$$x = \begin{cases} \frac{\tau_1(j)+(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta(k(j)-1) \\ \frac{-\tau_2(j)+(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta k(j) \end{cases}.$$

The algebra is analogous for the case where we measure incorrectly, so that

$$N_\theta(j+1) = \begin{cases} \frac{1+(\beta-\alpha)x}{2\beta}, & \text{with probability } q \\ \frac{1-(\beta-\alpha)x}{2\alpha}, & \text{with probability } p \end{cases},$$

where x is as defined above.

Before bounding $\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)]$, we prove three useful lemmas.

Lemma A.2. *For* $0 \leq a \leq 1$ *and* $0 \leq \tau \leq \frac{2}{m}$,

$$\left(\frac{\tau + (m-2)/m - a}{1-a}\right)\left(\frac{a-\tau}{a}\right) \leq \frac{m-2}{m}.$$

Proof. Rewrite the left hand side in two ways as follows.

$$\left(\frac{\tau + (m-2)/m - a}{1-a}\right)\left(\frac{a-\tau}{a}\right) = \tag{A.8}$$

$$\left(\frac{\tau - a + 1 - \frac{2}{m}}{1-a}\right)\left(\frac{a-\tau}{a}\right) \tag{A.9}$$

$$= \left(\frac{\tau - \frac{2}{m}}{1-a} + 1\right)\left(\frac{a-\tau}{a}\right). \tag{A.10}$$

Now consider two cases, $a \leq \tau$ and $a > \tau$.

Case $a \leq \tau$ In equation (A.9), the left term of the product is positive but the right term is nonpositive, making our whole term less than or equal to zero.

Case $a > \tau$ Note that both terms in the product of equation (A.10) are less than or equal to one because $0 \leq \tau \leq \frac{2}{m}$. First consider the situation where $\frac{\tau}{a} \geq \frac{2}{m}$. Then

$$\left(\frac{\tau - \frac{2}{m}}{1-a} + 1\right)\left(\frac{a-\tau}{a}\right) \leq 1 - \frac{\tau}{a} \leq \frac{m-2}{m}.$$

Next consider the situation when $\frac{\tau}{a} < \frac{2}{m}$. This implies that $\frac{m\tau}{2} < a$ and therefore that

$$\frac{\tau - \frac{2}{m}}{1-a} < \frac{\tau - \frac{2}{m}}{1 - \frac{m\tau}{2}} = \frac{2}{m}\left(\frac{m\tau - 2}{2 - m\tau}\right) = -\frac{2}{m}.$$

This implies the left hand term of the product in equation (A.10) is

$$\frac{\tau - \frac{2}{m}}{1-a} + 1 \leq \frac{m-2}{m},$$

making the product again less than $(m-2)/m$, which completes the proof. □

Lemma A.3. *For all* $0 < a < 1$ *and* $\tau \leq \frac{2m-2}{m}$,

$$\frac{\tau - (m-2)/m - a}{1-a} \leq \tau + \frac{m-2}{m}.$$

For all $0 < a < 1$ *and* $\tau \geq -\frac{2}{m}$,

$$\frac{-\tau + (m-2)/m - a}{1-a} \leq -\tau + \frac{m-2}{m}.$$

Proof. We prove the first statement given in the lemma. The proof of the second statement is nearly identical. Note that the statement from the lemma is equivalent to

$$\begin{aligned}\tau - \frac{m-2}{m} - a &\leq \left(\tau + \frac{m-2}{m}\right)(1-a) \\ &= \tau + \frac{m-2}{m} - \tau a - \left(\frac{m-2}{m}\right) a.\end{aligned}$$

Equivalently, we rearrange further to see that the original statement holds if

$$-a \leq 2\left(\frac{m-2}{m}\right) - \tau a - \left(\frac{m-2}{m}\right) a,$$

and hence if

$$a\left(\frac{m-2}{m} + \tau - 1\right) \leq 2\left(\frac{m-2}{m}\right).$$

Note since $a < 1$

$$\begin{aligned}a\left(\frac{m-2}{m} + \tau - 1\right) &\leq \frac{m-2}{m} + \tau - 1 \\ &\leq 2\left(\frac{m-2}{m}\right),\end{aligned}$$

completing the proof of the first statement. □

We are now ready to bound $\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)]$. From the definitions given in the algorithm description, it is straightforward to see that $0 \leq \tau_1(j) \leq \frac{2}{m}$, $0 < \tau_2(j) \leq \frac{2m-2}{m}$, and $\tau_1(j) + \tau_2(j) = 2a_{k(j)}(j) = 2a_{k(\theta)}(j)$. Also define

$$\begin{aligned}g(x) &= \frac{q}{2\beta}(1 + (\beta - \alpha)x) + \frac{p}{2\alpha}(1 - (\beta - \alpha)x) \\ &= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)x,\end{aligned}$$

so that $\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)] = P_1(j)g(x) + P_2(j)g(x)$, where x is defined above for the various cases and $P_1(j)$ and $P_2(j)$ are defined in Algorithm A.2. Consider case (i), $k(j) = k(\theta)$. Then applying Lemma 2 and rearranging terms shows

$$\begin{aligned}
\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)] &= P_1(j)g\left(\tfrac{\tau_1(j)+(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right) + P_2(j)g\left(\tfrac{\tau_2(j)-(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right) \\
&= \tfrac{q}{2\beta} + \tfrac{p}{2\alpha} + \left(\tfrac{q}{2\beta} - \tfrac{p}{2\alpha}\right)(\beta-\alpha) \times \\
&\quad \left[P_1(j)\left(\tfrac{\tau_1(j)+(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right) + P_2(j)\left(\tfrac{\tau_2(j)-(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right)\right] \\
&= \tfrac{q}{2\beta} + \tfrac{p}{2\alpha} + \left(\tfrac{q}{2\beta} - \tfrac{p}{2\alpha}\right)(\beta-\alpha) \times \\
&\quad \left[P_1(j)\left(\tfrac{\tau_1(j)+(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right) + P_2(j)\left(\tfrac{-\tau_1(j)-(m-2)/m+a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right)\right] \\
&= \tfrac{q}{2\beta} + \tfrac{p}{2\alpha} + \left(\tfrac{q}{2\beta} - \tfrac{p}{2\alpha}\right)(\beta-\alpha)\left(\tfrac{\tau_1(j)+(m-2)/m-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)}\right)(P_1(j)-P_2(j)).
\end{aligned}$$

Note that

$$\begin{aligned}
P_1(j) - P_2(j) &= \frac{\tau_2(j)-\tau_1(j)}{\tau_1(j)+\tau_2(j)} \\
&= \frac{a_{k(j)}(j)-\tau_1(j)}{a_{k(j)}(j)}.
\end{aligned}$$

Then by Lemma A.2, we have that

$$\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)] \le \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta-\alpha)\left(\frac{m-2}{m}\right).$$

Using Lemma A.3 for $\tau \le (2m-2)/m$, we see that for case (ii) ($k(j) < k(\theta)$)

$$\begin{aligned}
\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)] &\le P_1(j)g\left(-\tau_1(j) + \frac{m-2}{m}\right) + P_2(j)g\left(\tau_2(j) + \frac{m-2}{m}\right) \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta-\alpha)\times \\
&\quad \left[(-P_1\tau_1(j) + P_2\tau_2(j)) + \left(\frac{m-2}{m}\right)(P_1+P_2)\right] \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta-\alpha)\left[(-P_1\tau_1(j) + P_2\tau_2(j)) + \frac{m-2}{m}\right] \\
&\le \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta-\alpha)\left(\frac{m-2}{m}\right).
\end{aligned}$$

Similarly applying Lemma A.3 for $\tau \geq -2/m$ in case (iii) ($k(j) > k(\theta)$) we have

$$\begin{aligned}
\mathbb{E}[N_\theta(j+1)|\mathbf{a}(j)] &\leq P_1(j) g\left(\tau_1(j) + \frac{m-2}{m}\right) + P_2(j) g\left(-\tau_2(j) + \frac{m-2}{m}\right) \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha) \times \\
&\qquad \left[(P_1\tau_1(j) - P_2\tau_2(j)) + \left(\frac{m-2}{m}\right)(P_1 + P_2)\right] \\
&= \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\left[(P_1\tau_1(j) - P_2\tau_2(j)) + \frac{m-2}{m}\right] \\
&\leq \frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\left(\frac{m-2}{m}\right).
\end{aligned}$$

For $m = 2$, these reduce to the bound given in [6].

We now complete the proof by using the above result into (A.6) and (A.7). Plugging in to (A.6), we see that

$$\mathbb{P}\left(|\hat{\theta}_n - \theta| > \Delta\right) \leq \frac{1-\Delta}{\Delta}\left[\frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\left(\frac{m-2}{m}\right)\right]^n.$$

Next, we bound the expected error

$$\begin{aligned}
\mathbb{E}|\hat{\theta}_n - \theta| &= \int_0^\infty \mathbb{P}\left(|\hat{\theta}_n - \theta| > t\right) dt \\
&= \int_0^\Delta \mathbb{P}\left(|\hat{\theta}_n - \theta| > t\right) dt + \int_\Delta^1 \mathbb{P}\left(|\hat{\theta}_n - \theta| > t\right) dt \\
&\leq \Delta + (1-\Delta)\mathbb{P}\left(|\hat{\theta}_n - \theta| > \Delta\right) \\
&\leq \Delta + \frac{(1-\Delta)^2}{\Delta}\left[\frac{q}{2\beta} + \frac{p}{2\alpha} + \left(\frac{q}{2\beta} - \frac{p}{2\alpha}\right)(\beta - \alpha)\left(\frac{m-2}{m}\right)\right]^n,
\end{aligned}$$

and the result follows from the values for Δ, α, and β indicated in the theorem.

□

Lemma A.4. *The truncated PQS (TPQS) algorithm satisfies the statement of Thm. 2.3.*

Proof. The proof of Thm. 2.3 relies on upper bounding $N_\theta(j)$ as defined in (A.5) above. We show that the TPQS algorithm satisfies

$$\hat{N}_\theta(j) \leq N_\theta(j), \tag{A.11}$$

where $\hat{N}_\theta(j)$ is the analogous term with parameters defined by TPQS. First, note that any discretized quantile search with parameter ϕ and updates as given in Section II.C of Chapter 2 follows the

discretized update

$$a_i(j+1) = \begin{cases} \frac{2\beta}{1+(\tau-(1-2\phi))(\beta-\alpha)} a_i(j) & i \le k \\ \frac{2\alpha}{1+(\tau-(1-2\phi))(\beta-\alpha)} a_i(j) & i > k \end{cases}$$

if $Y_{j+1} = 0$ and

$$a_i(j+1) = \begin{cases} \frac{2\alpha}{1+(1-2\phi)-\tau)(\beta-\alpha)} a_i(j) & i \le k \\ \frac{2\beta}{1+(1-2\phi)-\tau)(\beta-\alpha)} a_i(j) & i > k \end{cases}$$

if $Y_{j+1} = 1$. Following the same algebra as in the PQS algorithm, we see that

$$\hat{N}_\theta(j+1) = \frac{1}{2\beta} + \frac{(\beta-\alpha)x}{2\beta},$$

with

$$x = \begin{cases} \frac{\tau_1(j)+(1-2\phi)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta(k(j)-1) \\ \frac{\tau_2(j)-(1-2\phi)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta k(j) \end{cases}$$

for case (i). Similarly, for cases (ii) and (iii), we have (respectively)

$$x = \begin{cases} \frac{-\tau_1(j)-(1-2\phi)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta(k(j)-1) \\ \frac{\tau_2(j)-(1-2\phi)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta k(j) \end{cases},$$

and

$$x = \begin{cases} \frac{\tau_1(j)+(1-2\phi)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta(k(j)-1) \\ \frac{-\tau_2(j)+(1-2\phi)-a_{k(\theta)}(j)}{1-a_{k(\theta)}(j)} & \text{if } X_{j+1} = \Delta k(j) \end{cases}.$$

We prove (A.11) in the case where $X_{j+1} = \tilde{X}_0$. The proof for the case where $X_{j+1} = \tilde{X}_1$ is symmetric. For the cases above where the $1-2\phi$ term has a positive coefficient (e.g., case (iii)), it is sufficient to show that

$$\phi \ge \frac{1}{m}.$$

Since TPQS chooses a point closer to the median than PQS, the above holds by definition. For the cases where the $1-2\phi$ term is negative (e.g., case (i) with $X_{j+1} = \Delta k(j)$), we prove a lemma analogous to Lemma A.3.

Lemma A.5. *For all* $0 < a < 1$ *and* $\tau \le 2 - 2\phi$*,*

$$\frac{\tau - (1-2\phi) - a}{1-a} \le \tau + \frac{m-2}{m}.$$

For all $0 < a < 1$ *and* $\tau \geq -2\phi$*,*

$$\frac{-\tau + (1 - 2\phi) - a}{1 - a} \leq -\tau + \frac{m-2}{m}.$$

Proof. We prove the first statement given in the lemma. The proof of the second statement is nearly identical. Note that the statement from the lemma is equivalent to

$$\begin{aligned}\tau - (1 - 2\phi) - a &\leq \left(\tau + \frac{m-2}{m}\right)(1 - a) \\ &= \tau + \frac{m-2}{m} - \tau a - \left(\frac{m-2}{m}\right) a.\end{aligned}$$

Equivalently, we rearrange further to see that the original statement holds if

$$-a \leq \frac{m-2}{m} + (1 - 2\phi) - \tau a - \left(\frac{m-2}{m}\right) a,$$

and hence if

$$a\left(\frac{m-2}{m} + \tau - 1\right) \leq 2\left(1 - \frac{1}{m} - \phi\right).$$

Note since $a < 1$

$$\begin{aligned}a\left(\frac{m-2}{m} + \tau - 1\right) &\leq \frac{m-2}{m} + \tau - 1 \\ &\leq 2\left(1 - \frac{1}{m} - \phi\right),\end{aligned}$$

completing the proof of the first statement. □

Using this lemma, the remainder of the proof of Thm. 2.3 follows directly. □

APPENDIX B

Proofs for SUPERPAC

In this Appendix, we provide the proofs to Theorem 3.1 and Corollary 3.1, which appear in Chapter 3.

Theorem B.1. *Consider two d-dimensional subspaces $\mathcal{S}_1$ and $\mathcal{S}_2$. Let $y = x + n$, where $x \in \mathcal{S}_1$ and $n \sim \mathcal{N}(0, \sigma^2 I_D)$. Define*

$$\mu(y) = \frac{\operatorname{dist}(y, \mathcal{S}_1)}{\operatorname{dist}(y, \mathcal{S}_2)} \, .$$

Then

$$\frac{(1-\varepsilon)\sqrt{\sigma^2(D-d)}}{(1+\varepsilon)\sqrt{\sigma^2(D-d) + \operatorname{dist}(x, \mathcal{S}_2)^2}} \leq \mu(y) \leq \frac{(1+\varepsilon)\sqrt{\sigma^2(D-d)}}{(1-\varepsilon)\sqrt{\sigma^2(D-d) + \operatorname{dist}(x, \mathcal{S}_2)^2}},$$

with probability at least $1 - 4e^{-c\varepsilon^2(D-d)}$, where c is an absolute constant.

Proof. The proof relies on theorem 5.2.1 from [39], restated below.

Theorem B.2. *(Concentration on Gauss space) Consider a random vector $X \sim \mathcal{N}(0, \sigma^2 I_D)$ and a Lipschitz function $f : \mathbb{R}^D \to \mathbb{R}$. Then for every $t \geq 0$,*

$$\mathbb{P}\left\{|f(X) - \mathbb{E}f(X)| \geq t\right\} \leq 2\exp\left(-\frac{ct^2}{\sigma^2 \left\|f\right\|_{\text{Lip}}^2}\right),$$

where $\left\|f\right\|_{\text{Lip}}$ is the Lipschitz constant of f.

First consider the numerator and note that $y - P_1 y = P_1^{\perp} y \sim \mathcal{N}(0, \sigma^2 P_1^{\perp})$ with

$$\mathbb{E}\left\|P_1^{\perp} y\right\|^2 = \sigma^2(D-d).$$

Let $f(z) = \|Pz\|_2$, where P is an arbitrary projection matrix. In this case, $\left\|f\right\|_{\text{Lip}} = 1$, as f is a composition of 1-Lipschitz functions, which is also 1-Lipschitz. Further, by Exercise 5.2.5 of [39],

we can replace $\mathbb{E}\|X\|_2$ by $\left(\mathbb{E}\|X\|_2^2\right)^{1/2}$ in the concentration inequality. Applying Thm. B.2 to the above, we see that

$$\mathbb{P}\left\{\left|\left\|P_1^{\perp}y\right\| - \sqrt{\sigma^2(D-d)}\right| \geq t\right\} \leq 2\exp\left(-\frac{ct^2}{\sigma^2}\right). \tag{B.1}$$

Similarly, for the denominator, note that $y - P_2 y = P_2^{\perp}y \sim \mathcal{N}(P_2^{\perp}x, \sigma^2 P_2^{\perp})$ with

$$\mathbb{E}\left\|P_2^{\perp}y\right\|^2 = \sigma^2(D-d) + \gamma^2.$$

Since $P_2^{\perp}y$ is no longer centered, we let $g(z) = z + P_2^{\perp}x$, which also has $\|g\|_{\text{Lip}} = 1$. Applying Thm. B.2 to the centered random vector $\bar{y} \sim \mathcal{N}(0, \sigma^2 P_2^{\perp})$ with Lipschitz function $h = f \circ g$, we have that

$$\mathbb{P}\left\{\left|\left\|P_2^{\perp}y\right\| - \sqrt{\sigma^2(D-d)+\gamma^2}\right| \geq t\right\} \leq 2\exp\left(-\frac{ct^2}{\sigma^2}\right). \tag{B.2}$$

Letting $t = \varepsilon\sqrt{\sigma^2(D-d)}$ in (B.1) and $t = \varepsilon\sqrt{\sigma^2(D-d)+\gamma^2}$ in (B.2) yields

$$(1-\varepsilon)\sqrt{\sigma^2(D-d)} \leq \left\|P_1^{\perp}y\right\| \leq (1+\varepsilon)\sqrt{\sigma^2(D-d)}$$

and

$$\begin{aligned}(1-\varepsilon)\sqrt{\sigma^2(D-d)+\gamma^2} &\leq \left\|P_2^{\perp}y\right\| \\ &\leq (1+\varepsilon)\sqrt{\sigma^2(D-d)+\gamma^2},\end{aligned}$$

each with probability at least $1 - 2\exp\left(-c\varepsilon^2(D-d)\right)$ (since $\gamma > 0$). Applying the union bound gives the statement of the theorem. □

Corollary B.1. *Suppose $x_1 \in \mathcal{S}_1$ is such that*

$$\operatorname{dist}(x_1, \mathcal{S}_2)^2 = \sin^2(\phi_1) + \delta\left(\frac{1}{d}\sum_{i=1}^{d}\sin^2(\phi_i)\right) \tag{B.3}$$

for some small $\delta \geq 0$; that is, x_1 is close to the intersection of $\mathcal{S}_1$ and $\mathcal{S}_2$. Let x_2 be a random point in $\mathcal{S}_1$ generated as $x_2 = U_1 w$ where U_1 is a basis for $\mathcal{S}_1$ and $w \sim \mathcal{N}(0, \frac{1}{d}I_d)$. We observe $y_i = x_i + n_i$, where $n_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, 2$. If there exists $\tau > 1$ such that

$$\delta < \frac{5}{7} - \frac{1}{\tau}$$

and

$$\tau\left(\sin^2(\phi_1)+\frac{1}{6}\sigma^2\left(D-d\right)\right)<\frac{1}{d}\sum_{i=1}^{d}\sin^2(\phi_i)\ , \tag{B.4}$$

that is, the average angle is sufficiently larger than the smallest angle, then

$$\mathbb{P}\left\{\mu(y_1)>\mu(y_2)\right\}\geq 1-e^{-c\left(\frac{7}{100}\right)^2 ds}-4e^{-c\left(\frac{1}{50}\right)^2(D-d)}$$

where $\mu(y)$ is defined as in Thm. B.1, c is an absolute constant, and $s=\frac{1}{d}\sum_{i=1}^{d}\sin^2(\phi_i)$.

Proof. We have from Thm. B.1 that

$$\mu(y_2)\leq\frac{(1+\varepsilon)\sqrt{\sigma^2(D-d)}}{(1-\varepsilon)\sqrt{\sigma^2(D-d)+\gamma_2^2}}$$

and

$$\frac{(1-\varepsilon)\sqrt{\sigma^2(D-d)}}{(1+\varepsilon)\sqrt{\sigma^2(D-d)+\gamma_1^2}}\leq\mu(y_1)$$

with probability at least $1-4e^{-c\varepsilon^2(D-d)}$. Therefore if we get the upper bound of $\mu(y_2)$ to be smaller than the lower bound of $\mu(y_1)$, we are done. Rearranging this desired inequality we see that we need

$$\gamma_1^2<\beta^4\gamma_2^2-(1-\beta^4)\sigma^2(D-d). \tag{B.5}$$

where $\beta=(1-\varepsilon)/(1+\varepsilon)$. Let ε be such that $\beta^4=5/6$, and let $\gamma_1^2=\sin^2(\phi_1)+\delta s$ as in the theorem. Then we wish to select δ to satisfy

$$\delta<\frac{\frac{5}{6}\gamma_2^2-\sin^2(\phi_1)-\frac{1}{6}\sigma^2(D-d)}{s}. \tag{B.6}$$

Applying concentration with γ_2^2, we have that $\gamma_2^2\geq(1-\xi)^2 s$ with probability at least $1-e^{-c\xi^2 ds}$ where c is an absolute constant. Therefore taking ξ to be such that $(1-\xi)^2=6/7$, we require

$$\delta<\frac{\frac{5}{7}s-\sin^2(\phi_1)-\frac{1}{6}\sigma^2(D-d)}{s}=\frac{5}{7}-\frac{1}{\tau}$$

where we used the definition of τ in the theorem. To quantify the probability we need the appropriate values for ε and ξ; we lower bound both with simple fractions: $1/50<\varepsilon$ where $\left((1-\varepsilon)/(1+\varepsilon)\right)^4=\beta=5/6$ and $7/100<\xi$ where $(1-\xi)^2=6/7$. Applying the union bound with the chosen concentration values implies that $\mu(y_1)>\mu(y_2)$ holds with probability at least $1-e^{-c\left(\frac{7}{100}\right)^2 ds}-4e^{-c\left(\frac{1}{50}\right)^2(D-d)}$. □

APPENDIX C

Proofs for EKSS

In this document, we provide the proof to Lemma 4.1 of Chapter 4, which appears in the proofs of both theorems of the chapter. We also state and prove a claim made in Section 4.1 about K-Subspaces.

C.1 Proof of Lemma 4.1

In this section, we prove Lemma 4.1 of Chapter 4, which allows us to leverage the connectivity results for TSC [41] in Theorems 4.1 and 4.2.

Lemma C.1. *The probability that two points $x_i, x_j \in \mathcal{X}$ are clustered together by one iteration of EKSS-0 (i.e., EKSS-0 with $B = 1$) is an increasing function of $\left|x_i^T x_j\right|$.*

Proof. Let $u, v \sim \mathcal{N}(0, \frac{1}{D} I_D)$ be two one-dimensional Gaussian candidates in EKSS-0. Then u and v are orthogonally invariant, i.e., $Qu, Qv \sim \mathcal{N}(0, \frac{1}{D} I_D)$ for any orthogonal matrix Q. Furthermore, x and y cluster together if and only if x and $-y$ cluster together. As a result, we can consider $x = [1\ 0\ \cdots\ 0]^T$ and $y = [\cos\theta\ \sin\theta\ 0\ \cdots 0]^T$ where $0 < \theta < \pi/2$ without loss of generality. The points x and y are clustered together when

$$(|u_1| > |v_1| \text{ and } |u_1 \cos\theta + \ u_2 \sin\theta| > |v_1 \cos\theta + \ v_2 \sin\theta|) \tag{C.1}$$

or

$$(|u_1| < |v_1| \text{ and } |u_1 \cos\theta + \ u_2 \sin\theta| < |v_1 \cos\theta + \ v_2 \sin\theta|). \tag{C.2}$$

Note that (C.1) and (C.2) are disjoint events and occur with equal probability since u and v are identically distributed. The remainder of the proof is dedicated to showing that the conditional probability of (C.1) given u_1 and v_1 is a decreasing function of θ when $|u_1| > |v_1|$. From this fact, it follows that the probability of (C.1) is a decreasing function of θ by the law of total probability (taken over u_1 and v_1). Lemma 4.1 follows then because (C.2) has equal probability and $\theta = \arccos(|x^T y|)$ is a decreasing function of $|x^T y|$.

Observe that

$$|u_1| > |v_1| \iff \Big[\underbrace{(u_1 + v_1 > 0 \text{ and } u_1 - v_1 > 0)}_{\text{i.e., } u_1 > |v_1|} \text{ or } \underbrace{(u_1 + v_1 < 0 \text{ and } u_1 - v_1 < 0)}_{\text{i.e., } u_1 < -|v_1|}\Big] \tag{C.3}$$

and likewise

$$|u_1 \cos\theta + u_2 \sin\theta| > |v_1 \cos\theta + v_2 \sin\theta| \tag{C.4}$$

$$\iff$$

$$(u_2 + v_2 > -(u_1 + v_1)\cot\theta \text{ and } u_2 - v_2 > -(u_1 - v_1)\cot\theta) \tag{C.5}$$

$$\text{or } (u_2 + v_2 < -(u_1 + v_1)\cot\theta \text{ and } u_2 - v_2 < -(u_1 - v_1)\cot\theta). \tag{C.6}$$

For convenience, let $s_1 = u_1 + v_1$, $d_1 = u_1 - v_1$, $s_2 = u_2 + v_2$ and $d_2 = u_2 - v_2$. The random variables s_2 and d_2 are i.i.d. Gaussian random variables because the vector $[s_2\ d_2]^T$ is a scaled rotation of $[u_2\ v_2]^T$ and u_2 and v_2 are i.i.d. Gaussian random variables. From (C.3) and (C.4), it follows that when $|u_1| > |v_1|$, either $s_1, d_1 > 0$ or $s_1, d_1 < 0$ and the conditional probability of (C.1) given u_1 and v_1 is

$$\begin{aligned}
\rho(\theta) &= \mathbb{P}\left\{|u_1| > |v_1| \text{ and } |u_1\cos\theta + u_2\sin\theta| > |v_1\cos\theta + v_2\sin\theta| : u_1, v_1\right\} \\
&= \mathbb{P}\left\{|u_1\cos\theta + u_2\sin\theta| > |v_1\cos\theta + v_2\sin\theta| : u_1, v_1\right\} \\
&= \mathbb{P}\left\{(s_2 > -s_1\cot\theta \text{ and } d_2 > -d_1\cot\theta) \text{ or } (s_2 < -s_1\cot\theta \text{ and } d_2 < -d_1\cot\theta) : s_1, d_1\right\} \\
&= \mathbb{P}\left\{s_2 > -s_1\cot\theta : s_1\right\}\mathbb{P}\left\{d_2 > -d_1\cot\theta : d_1\right\} + \mathbb{P}\left\{s_2 < -s_1\cot\theta : s_1\right\} \times \\
&\quad \mathbb{P}\left\{d_2 < -d_1\cot\theta : d_1\right\} \\
&= (1 - F(-s_1\cot\theta))(1 - F(-d_1\cot\theta)) + F(-s_1\cot\theta)F(-d_1\cot\theta)
\end{aligned} \tag{C.7}$$

where $F(x) = \int_{-\infty}^{x} f(\tau)d\tau$ is the CDF for the i.i.d. Gaussian random variables u_2 and v_2 with density $f(x)$.

Differentiating (C.7) with respect to θ and factoring yields

$$\rho'(\theta) = -\underbrace{\csc^2\theta}_{>0}\left[\underbrace{f(-d_1\cot\theta)}_{>0}\underbrace{d_1(1 - 2F(-s_1\cot\theta))}_{=:\delta_1} + \underbrace{f(-s_1\cot\theta)}_{>0}\underbrace{s_1(1 - 2F(-d_1\cot\theta))}_{=:\delta_2}\right]$$

Recall that either $s_1, d_1 > 0$ or $s_1, d_1 < 0$. We consider each case:

1. If $s_1, d_1 > 0$ then $F(-s_1\cot\theta), F(-d_1\cot\theta) \leq 1/2$ with equality only when $\theta = \pi/2$. As a result $\delta_1, \delta_2 \geq 0$ with equality only when $\theta = \pi/2$.

2. If $s_1, d_1 < 0$ then $F(-s_1 \cot\theta), F(-d_1 \cot\theta) \geq 1/2$ with equality only when $\theta = \pi/2$. Once again $\delta_1, \delta_2 \geq 0$ with equality only when $\theta = \pi/2$.

Thus it follows that $\rho'(\theta) \leq 0$ with equality only when $\theta = \pi/2$ and so the conditional probability $\rho(\theta)$ of (C.1) given u_1 and v_1 is a decreasing function of θ when $|u_1| > |v_1|$. □

C.2 Non-Global Convergence of K-subspaces

In this section, we prove the claim that there is a set of initializations of nonzero measure that will necessarily lead the K-subspaces (KSS) algorithm to a solution that is not a global minimizer for the simple case of two one-dimensional subspaces of $\mathbb{R}^2$.

Proposition C.1. *Consider two one-dimensional subspaces $\mathcal{S}_1, \mathcal{S}_2 \subset \mathbb{R}^2$ having angle between them $\theta \in (0, \pi/2)$. The set of initializations such that KSS converges to a clustering that is not a global optimum has nonzero measure.*

Proof. Let u_1, u_2 be the true subspace bases, and let v_1, v_2 be the candidate bases initialized uniformly at random from the unit sphere. Let $\theta_{u,v}$ denote the angle between vectors u and v, i.e., $\theta_{u,v} = \cos^{-1}\left(u^T v\right)$, where by symmetry we only consider angles at most $\pi/2$. By rotation invariance of the candidate subspaces, we assume that $u_1 = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$ and $u_2 = \begin{bmatrix} \cos(\theta) & \sin(\theta) \end{bmatrix}^T$ without loss of generality. Consider the case where $\theta_{u_1,v_1} > \theta$ and $\theta_{u_1,v_2} > \theta$, and note that each event occurs independently with probability $\pi/2 - \theta$. Next, note that $\theta_{u_2,v_1} < \theta_{u_2,v_2}$ implies $\theta_{u_1,v_1} < \theta_{u_1,v_2}$, in which case all points are assigned to v_1. Similarly, $\theta_{u_2,v_2} < \theta_{u_2,v_1}$ implies $\theta_{u_1,v_2} < \theta_{u_1,v_1}$, in which case all points are assigned to v_2. Thus, all points are clustered to the same candidate subspace as long as both $\theta_{u_1,v_1} > \theta$ and $\theta_{u_1,v_2} > \theta$, which occurs with probability $(\pi/2 - \theta)^2$. Under this event, KSS converges after the first iteration, as the subspace corresponding one candidate is null. Hence, the set of initializations such that KSS converges to a local and not global optimum has measure at least $(\pi/2 - \theta)^2$. □

BIBLIOGRAPHY

[1] C. Xiong, D. M. Johnson, and J. J. Corso, "Active clustering with model-based uncertainty reduction," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 1, pp. 5–17, 2017.

[2] P. Donmez and J. G. Carbonell, "Proactive learning: cost-sensitive active learning with multiple imperfect oracles," in *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 2008, pp. 619–628.

[3] G. L. E. R. Laboratory, "Lake Erie hypoxia warning system," http://www.glerl.noaa.gov/res/waterQuality/, 2005.

[4] D. Beletsky, D. Schwab, and M. McCormick, "Modeling the 1998-2003 summer circulation and thermal structure in Lake Michigan," *Journal of Geophys. Res.*, vol. 111, 2006.

[5] A. Singh, R. Nowak, and P. Ramanathan, "Active learning for adaptive mobile sensing networks," in *Proc. Information Processing in Sensor Networks*, 2006.

[6] R. Castro and R. Nowak, "Active learning and sampling," in *Foundations and Applications of Sensor Management*, 1st ed. New York, NY: Springer, 2008, ch. 8.

[7] R. Basri and D. Jacobs, "Lambertian reflectance and linear subspaces," *IEEE TPAMI*, vol. 25, no. 2, pp. 218–233, February 2003.

[8] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database of handwritten digits," 2016. [Online]. Available: yann.lecun.com/exdb/mnist

[9] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized non-negative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[10] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (COIL-20)," Columbia University, Tech. Rep., 1996.

[11] ——, "Columbia object image library (COIL-100)," Columbia University, Tech. Rep., 1996.

[12] I. Davidson, K. L. Wagstaff, and S. Basu, "Measuring constraint-set utility for partitional clustering algorithms," in *Proc. European Conf. on Machine Learning and Prinicpals and Practice of Knowledge Discovery in Databases*, 2006.

[13] P. S. Bradley and O. L. Mangasarian, "k-Plane clustering," *Journal of Global Optimization*, vol. 16, pp. 23–32, 2000.

[14] P. Tseng, "Nearest q-flat to m points," *Journal of Optimization Theory and Applications*, vol. 105, no. 1, pp. 249–252, 2000.

[15] P. K. Agarwal and N. H. Mustafa, "K-means projective clustering," in *Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*. ACM, 2004, pp. 155–165.

[16] A. L. Fred and A. K. Jain, "Combining multiple clusterings using evidence accumulation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 27, no. 6, pp. 835–850, 2005.

[17] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[18] Y. Freund, R. E. Schapire *et al.*, "Experiments with a new boosting algorithm," in *ICML*, vol. 96, 1996, pp. 148–156.

[19] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.

[20] D. H. Wolpert, "The lack of a priori distinctions between learning algorithms," *Neural computation*, vol. 8, no. 7, pp. 1341–1390, 1996.

[21] J. C. Dunn, "Well-separated clusters and optimal fuzzy partitions," *Journal of cybernetics*, vol. 4, no. 1, pp. 95–104, 1974.

[22] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[23] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, pp. 2765–2781, Nov. 2013.

[24] J. Lipor, D. Hong, D. Zhang, and L. Balzano, "Subspace clustering using ensembles of k-subspaces," *arXiv preprint arXiv:1709.04744*, 2017.

[25] S. Ben-David and M. Ackerman, "Measures of clustering quality: A working set of axioms for clustering," in *Advances in neural information processing systems*, 2009, pp. 121–128.

[26] J. Lipor and L. Balzano, "Quantile search: A distance-penalized active learning algorithm for spatial sampling," in *Proc. Allerton Conf. on Communication, Control, and Computing*, 2015.

[27] J. Lipor, B. P. Wong, D. Scavia, B. Kerkez, and L. Balzano, "Distance-penalized active learning using quantile search," *IEEE Transactions on Signal Processing*, vol. 65, no. 20, Oct. 2017, accepted for publication.

[28] J. Lipor and L. Balzano, "Margin-based active subspace clustering," in *Proc. Int. Workshop on Computational Advances in Multi-Sensor Adaptive Processing*, 2015.

[29] ——, "Leveraging union of subspace structure to improve constrained clustering," in *Proc. Int. Conf. on Machine Learning*, 2017.

[30] ——, "Robust blind calibration via total least squares," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, 2014.

[31] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of machine learning*. MIT press, 2012.

[32] R. Castro, R. Willett, and R. Nowak, "Faster rates in regression via active learning," in *Proc. Advances in Neural Information Processing Systems*, 2005.

[33] Y. Wang and A. Singh, "Noise-adaptive margin-based active learning for multi-dimensional data and lower bounds under tsybakov noise," in *Proc. AAAI Conference on Artificial Intellgence*, 2016.

[34] S. Dasgupta, "Analysis of a greedy active learning strategy," in *Proc. Advances in Neural Information Processing Systems*, 2005.

[35] R. Nowak, "Generalized binary search," in *Proc. Allerton Conference on Communication, Control, and Computing*, 2008.

[36] ——, "The geometry of generalized binary search," *IEEE Trans. Inf. Theory*, vol. 57, pp. 7893–7906, 2011.

[37] B. Settles, *Active Learning*. Morgan & Claypool, 2012.

[38] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.

[39] R. Vershynin, *A Course in High Dimensional Probability*, 2017. [Online]. Available: www-personal.umich.edu/~romanv/teaching/2015-16/626/HDP-book.pdf

[40] D. Park, C. Caramanis, and S. Sanghavi, "Greedy subspace clustering," in *Advances in Neural Information Processing Systems*, 2014, pp. 2753–2761.

[41] R. Heckel and H. Bölcskei, "Robust subspace clustering via thresholding," *IEEE Trans. Inf. Theory*, vol. 24, no. 11, pp. 6320–6342, 2015.

[42] G. Golub and C. V. Loan, *Matrix Computations*. Johns Hopkins University Press, 2012.

[43] M. Soltanolkotabi and E. J. Candes, "A Geometric Analysis of Subspace Clustering with Outliers," *The Annals of Statistics*, vol. 40, no. 4, pp. 2195–2238, 2012.

[44] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[45] D. Scavia, J. D. Allan, K. K. Arend, S. Bartell, D. Beletsky, N. S. Bosch, S. B. Brandt, R. D. Briland, I. Daloğlu, J. V. DePinto *et al.*, "Assessing and addressing the re-eutrophication of lake erie: Central basin hypoxia," *Journal of Great Lakes Research*, vol. 40, no. 2, pp. 226–246, 2014.

[46] R. Castro and R. Nowak, "Minimax bounds for active learning," *IEEE Trans. Inf. Theory*, vol. 54, pp. 2339–2353, May 2008.

[47] N. Oceanic and A. Administration, "Bathymetry of lake erie & lake saint clair," http://www.ngdc.noaa.gov/mgg/greatlakes/erie.html, 2015.

[48] M. Horstein, "Sequential decoding using noiseless feedback," *IEEE Trans. Inf. Theory*, vol. 9, 1963.

[49] M. V. Burnashev and K. S. Zigangirov, "An interval estimation problem for controlled observations," *Problems in Information Transmission*, vol. 10:223-231, 1974, translated from Problemy Peredachi Informatsii, 10(3):51-61, July-September, 1974.

[50] R. M. Karp and R. Kleinberg, "Noisy binary search and its applications," in *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 2007.

[51] M. B. Or and A. Hassidim, "The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well)," in *Proc. IEEE Symposium of Foundations of Computer Science*, 2008.

[52] R. Waeber, P. I. Frazier, and S. G. Henderson, "Bisection search with noisy responses," *SIAM Journal on Control and Optimization*, vol. 51, pp. 2261–2279, 2013.

[53] A. Ramdas and A. Singh, "Algorithmic connections between active learning and stochastic convex optimization," in *International Conference on Algorithmic Learning Theory*. Springer, 2013, pp. 339–353.

[54] T. Tsiligkaridis, "Asynchronous decentralized algorithms for the noisy 20 questions problem," in *2016 IEEE International Symposium on Information Theory (ISIT)*. IEEE, 2016, pp. 2699–2703.

[55] A. Liu, G. Jun, and J. Ghosh, "Spatially cost-sensitive active learning," in *Proc. SIAM Conf. on Data Mining*, 2009.

[56] B. Demir, L. Minello, and L. Bruzzone, "A cost-sensitive active learning technique for the definition of effective training sets for supervised classifiers," in *Proc. SIAM Conf. on Data Mining*, 2009.

[57] A. Guillory and J. Blimes, "Average-case active learning with costs," in *Proc. Algorithmic Learning Theory*, 2009.

[58] J. Unnikrishnan and M. Vetterli, "Sampling and reconstruction of spatial fields using mobile sensors," *IEEE Trans. Sig. Proc*, vol. 61, pp. 2328–2340, 2013.

[59] ——, "Sampling high-dimensional bandlimited fields on low-dimensional manifolds," *IEEE Trans. Inf. Theory*, vol. 59, pp. 2103–2127, 2013.

[60] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser, "Efficient informative sensing using multiple robots," *Journal of Artificial Intelligence Research*, vol. 34, pp. 707–755, 2009.

[61] A. Krause, A. Singh, and C. Guestrin, "Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies," *Journal of Machine Learning Research*, vol. 9, no. Feb, pp. 235–284, 2008.

[62] H. Bayrum, J. V. Hook, and V. Isler, "Gathering bearing data for target localization," *IEEE Robotics and Automation Letters*, vol. 1, no. 1, pp. 369–374, Jan. 2016.

[63] M. Rahimi, M. Hansen, W. J. Kaiser, G. S. Sukhatme, and D. Estrin, "Adaptive sampling for environmental field estimation using robotic sensors," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2005, pp. 3692–3698.

[64] B. Schlegel, R. Gemulla, and W. Lehner, "k-ary search on modern processors," in *Proc. Fifth Int. Workshop on Data Management on New Hardware*, 2009.

[65] J. Jiang and K. R. Narayanan, "Multilevel coding for channels with non-uniform inputs and rateless transmission over the bsc," in *2006 IEEE International Symposium on Information Theory*. IEEE, 2006, pp. 518–522.

[66] Y. Zhou, D. R. Obenour, D. Scavia, T. H. Johengen, and A. M. Michalak, "Spatial and temporal trends in lake erie hypoxia, 1987–2007," *Environmental science & technology*, vol. 47, no. 2, pp. 899–905, 2013.

[67] T. Bridgeman, C. Wallace, G. Carter, R. Carvajal, L. Schiesari, S. Aslam, E. Cloyd, D. Elder, A. Field, K. Schulz *et al.*, "A limnological survey of third sister lake, michigan with historical comparisons," *Lake and Reservoir Management*, vol. 16, no. 4, pp. 253–267, 2000.

[68] A. Valada, P. Velagapudi, B. Kannan, C. Tomaszewski, G. Kantor, and P. Scerri, "Development of a low cost multi-robot autonomous marine surface platform," in *Field and Service Robotics*. Springer, 2014, pp. 643–658.

[69] B. P. Wong and B. Kerkez, "Real-time environmental sensor data: An application to water quality using web services," *Environmental Modelling & Software*, vol. 84, pp. 505–517, 2016.

[70] G. Dasarathy and R. Nowak, "$\mathcal{S}^2$: An efficient graph-based active learning algorithm with application to nonparametric learning," in *Proc. Conf. on Learning Theory*, 2015.

[71] Y. R. Wang and A. Singh, "Algorithmic connections between active learning and stochastic convex optimization," in *Proc. Algorithmic Learning Theory*, 2013.

[72] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 831–843, 2000.

[73] R. Vidal, S. S. Sastry, and Y. Ma, *Generalized Principal Component Analysis*. Springer-Verlag, 2016.

[74] A. Biswas and D. Jacobs, "Active image clustering with pairwise constraints from humans," *International Journal on Computer Vision*, vol. 108, pp. 133–147, 2014.

[75] R. Vidal, "Subspace clustering," *IEEE Signal Processing Magazine*, vol. 28, pp. 52–68, Mar. 2011.

[76] C. You, D. P. Robinson, and R. Vidal, "Scalable sparse subspace clustering by orthogonal matching pursuit," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[77] C. You, C.-G. Li, D. P. Robinson, and R. Vidal, "Oracle based active set algorithm for scalable elastic net subspace clustering," in *Proc. IEEE International Conference on Computer Vision and Pattern Recognition*, 2016.

[78] G. Liu, Z. Lin, and Y. Yu, "Robust subspace segmentation by low-rank representation," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 663–670.

[79] M. Soltanolkotabi and E. J. Candes, "Robust Subspace Clustering," *The Annals of Statistics*, vol. 42, no. 2, pp. 669–699, 2014.

[80] Y.-X. Wang and H. Xu, "Noisy sparse subspace clustering," in *Proceedings of The 30th International Conference on Machine Learning*, 2013, pp. 89–97.

[81] Y. Wang, Y.-X. Wang, and A. Singh, "Graph connectivity in noisy sparse subspace clustering," in *Proceedings of The 19th International Conference on Artificial Intelligence and Statistics*, 2016.

[82] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl, "Constrained K-means clustering with background knowledge," in *Proc. Int. Conf. on Machine Learning*, 2001.

[83] S. Basu, A. Banerjee, and R. J. Mooney, "Active semi-supervision for pairwise constrained clustering," in *Proc. SIAM Int. Conf. on Data Mining*, 2004.

[84] P. K. Mallapragada, R. Jin, and A. K. Jain, "Active query selection for semi-supervised clustering," in *Proc. Int. Conf. on Pattern Recognition*, 2008.

[85] Q. Xu, M. desJardins, and K. L. Wagstaff, "Active constrained clustering by examining spectral eigenvectors," in *Proc. 8th Int. Conf. on Discovery Science*, 2005.

[86] X. Wang and I. Davidson, "Active spectral clustering," in *Proc. 10th Int. Conf. on Data Mining*, 2010.

[87] C. Xiong, personal correspondence.

[88] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 2, pp. 210–227, 2009.

[89] J. Haupt, R. M. Castro, and R. Nowak, "Distilled sensing: Adaptive sampling for sparse detection and estimation," *IEEE Transactions on Information Theory*, vol. 57, no. 9, pp. 6222–6235, 2011.

[90] P. Indyk, E. Price, and D. P. Woodruff, "On the power of adaptivity in sparse recovery," in *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*. IEEE, 2011, pp. 285–294.

[91] A. Soni and J. Haupt, "On the fundamental limits of recovering tree sparse vectors from noisy linear measurements," *IEEE Transactions on Information Theory*, vol. 60, no. 1, pp. 133–149, 2014.

[92] A. Krishnamurthy and A. Singh, "Low-rank matrix and tensor completion via adaptive sampling," in *Advances in Neural Information Processing Systems*, 2013, pp. 836–844.

[93] E. J. Candes, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM*, vol. 58, no. 3, May 2011.

[94] Z. Lin, M. Chen, L. Wu, and Y. Ma, "Linearized alternating direction method with adaptive penalty for low-rank representation," in *Advances in Neural Information Processing Systems*, 2011.

[95] T. Zhang, A. Szlam, Y. Wang, and G. Lerman, "Hybrid linear modeling via local best-fit flats," *International Journal of Computer Vision*, vol. 100, pp. 217–240, 2012.

[96] L. A. Park, "Fast approximate text document clustering using compressive sampling," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 565–580.

[97] A. Ruta and F. Porikli, "Compressive clustering of high-dimensional data," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 1. IEEE, 2012, pp. 380–385.

[98] T. Wimalajeewa, H. Chen, and P. K. Varshney, "Performance limits of compressive sensing-based signal classification," *IEEE Transactions on Signal Processing*, vol. 60, no. 6, pp. 2758–2770, 2012.

[99] Y. Wang, Y.-X. Wang, and A. Singh, "A deterministic analysis of noisy sparse subspace clustering for dimensionality-reduced data," in *Proc. of Int. Conf. on Machine Learning (ICML)*, 2015, pp. 1422–1431.

[100] D. Pimentel-Alarcón, L. Balzano, and R. Nowak, "Necessary and sufficient conditions for sketched subspace clustering," in *Communication, Control, and Computing (Allerton), 2016 54th Annual Allerton Conference on*. IEEE, 2016, pp. 1335–1343.

[101] N. Halko, P. Martinsson, and J. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM Review*, vol. 53, no. 2, pp. 217–288, 2011.

[102] P. Drineas and M. W. Mahoney, "RandNLA: Randomized numerical linear algebra," *Communications of the ACM*, vol. 59, no. 6, pp. 80–90, 2016.

[103] R. Tron and R. Vidal, "A benchmark for the comparison of 3-D motion segmentation algorithms," in *IEEE Int. Conf. on Comp. Vision and Pattern Recog.*, 2011.

[104] A. Georghiades, P. Belhumeur, and D. Kriegman, "From few to many: Illumination cone models for face recognition under variable lighting and pose," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 23, no. 6, pp. 643–660, 2001.

[105] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, and S. Yan, "Robust and efficient subspace segmentation via least squares regression," *Computer Vision–ECCV 2012*, pp. 347–360, 2012.

[106] R. Vidal and P. Favaro, "Low rank subspace clustering (LRSC)," *Pattern Recognition Letters*, vol. 43, pp. 47–61, 2014.

[107] J. Shen, P. Li, and H. Xu, "Online low-rank subspace clustering by basis dictionary pursuit," in *Proc. International Conference on Machine Learning*, 2016.

[108] T. Zhang, A. Szlam, and G. Lerman, "Median k-flats for hybrid linear modeling with many outliers," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 234–241.

[109] L. Balzano, A. Szlam, B. Recht, and R. Nowak, "k-subspaces with missing data," in *Statistical Signal Processing Workshop (SSP), 2012 IEEE*. IEEE, 2012, pp. 612–615.

[110] J. He, Y. Zhang, J. Wang, N. Zeng, and H. Hao, "Robust k-subspaces recovery with combinatorial initialization," in *Big Data (Big Data), 2016 IEEE International Conference on*. IEEE, 2016, pp. 3573–3582.

[111] J. Ghosh and A. Acharya, "Cluster ensembles," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 4, pp. 305–315, 2011.

[112] F. Leisch, "Bagged clustering," *SFB Adaptive Information Systems and Modelling in Economics and Management Science, WU Vienna University of Economics and Business*, 1999.

[113] B. Minaei-Bidgoli, A. Topchy, and W. F. Punch, "Ensembles of partitions via data resampling," in *Information Technology: Coding and Computing, 2004. Proceedings. ITCC 2004. International Conference on*, vol. 2. IEEE, 2004, pp. 188–192.

[114] K. Tumer and A. K. Agogino, "Ensemble clustering with voting active clusters," *Pattern Recognition Letters*, vol. 29, no. 14, pp. 1947–1953, 2008.

[115] A. Topchy, A. K. Jain, and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 27, no. 12, pp. 1866–1881, 2005.

[116] A. Fred and A. K. Jain, "Evidence accumulation clustering based on the k-means algorithm," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2002, pp. 442–451.

[117] A. L. Fred and A. K. Jain, "Data clustering using evidence accumulation," in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, vol. 4. IEEE, 2002, pp. 276–280.

[118] S. R. Bulò, A. Lourenço, A. Fred, and M. Pelillo, "Pairwise probabilistic clustering using evidence accumulation," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. Springer, 2010, pp. 395–404.

[119] A. Lourenço, S. R. Bulò, N. Rebagliati, A. L. Fred, M. A. Figueiredo, and M. Pelillo, "Probabilistic evidence accumulation for clustering ensembles." in *ICPRAM*, 2013, pp. 58–67.

[120] ——, "Probabilistic consensus clustering using evidence accumulation," *Machine Learning*, vol. 98, no. 1-2, pp. 331–357, 2015.

[121] A. Ng, Y. Weiss, and M. Jordan, "On spectral clustering: Analysis and an algorithm," in *Proc. Neural Information Processing Systems*, 2001.

[122] D. Zhang and L. Balzano, "Global convergence of a grassmannian gradient descent algorithm for subspace estimation," in *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, A. Gretton and C. C. Robert, Eds., vol. 51. Cadiz, Spain: PMLR, 09–11 May 2016, pp. 1460–1468. [Online]. Available: http://proceedings.mlr.press/v51/zhang16b.html

[123] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[124] P. Jain, P. Netrapalli, and S. Sanghavi, "Low-rank matrix completion using alternating minimization," in *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*. ACM, 2013, pp. 665–674.

[125] C. Tomasi and T. Kanade, "Shape and motion from image streams under orthography," *Int'l J. Computer Vision*, vol. 9, no. 2, pp. 137–154, 1992.

[126] R. Heckel, E. Agustsson, and H. Bolcskei, "Neighborhood selection for thresholding-based subspace clustering," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 6761–6765.

[127] K. Lee, J. Ho, and D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intelligence*, vol. 27, no. 5, pp. 684–698, 2005.

[128] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1872–1886, 2013.

[129] M. Rahmani and G. Atia, "Coherence pursuit: Fast, simple, and robust principal component analysis," *arXiv preprint arXiv:1609.04789*, 2016.

[130] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[131] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, "Understanding of internal clustering validation measures," in *Data Mining (ICDM), 2010 IEEE 10th International Conference on*. IEEE, 2010, pp. 911–916.

[132] I. Guyon, U. Von Luxburg, and R. C. Williamson, "Clustering: Science or art," in *NIPS 2009 workshop on clustering theory*, 2009, pp. 1–11.

[133] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.

[134] M. Ackerman and S. Ben-David, "A characterization of linkage-based hierarchical clustering," *Journal of Machine Learning Research*, 2013.

[135] T. Van Laarhoven and E. Marchiori, "Axioms for graph clustering quality functions." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 193–215, 2014.

[136] A. Biswas and B. Biswas, "Defining quality metrics for graph clustering evaluation," *Expert Systems with Applications*, vol. 71, pp. 1–17, 2017.

[137] J. Friedman, T. Hastie, and R. Tibshirani, *The elements of statistical learning*. Springer series in statistics New York, 2001, vol. 1.

[138] H. Almeida, D. Guedes, W. Meira, and M. J. Zaki, "Is there a best quality metric for graph clusters?" in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2011, pp. 44–59.

[139] J. Creusefond, T. Largillier, and S. Peyronnet, "On the evaluation potential of quality functions in community detection for different contexts," in *International Conference and School on Network Science*. Springer, 2016, pp. 111–125.

[140] T. Chakraborty, A. Dalmia, A. Mukherjee, and N. Ganguly, "Metrics for community analysis: A survey," *arXiv preprint arXiv:1604.03512*, 2016.

[141] U. Brandes, M. Gaertler, and D. Wagner, *Experiments on graph clustering algorithms*. Springer, 2003.

[142] M. E. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical review E*, vol. 69, no. 2, p. 026113, 2004.

[143] J. M. Kleinberg, "An impossibility theorem for clustering," in *Advances in neural information processing systems*, 2003, pp. 463–470.

[144] R. B. Zadeh and S. Ben-David, "A uniqueness theorem for clustering," in *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press, 2009, pp. 639–646.

[145] R. Kłopotek and M. Kłopotek, "On the discrepancy between kleinberg's clustering axioms and k-means clustering algorithm behavior," *arXiv preprint arXiv:1702.04577*, 2017.

[146] S. Dasgupta, C. H. Papadimitriou, and U. Vazirani, *Algorithms*. McGraw-Hill, Inc., 2006.

[147] D. Golovin and A. Krause, "Adaptive submodularity: Theory and applications in active learning and stochastic optimization," *Journal of Artificial Intelligence Research*, vol. 42, pp. 427–486, 2011.